Rosenberg discuss five principles of cloud computing, i.e.:

* Pooled Computing Resources
* Virtualized Computing Resources
* Elastic Scaling up or down
* Automatic creation and deletion of new VM's
* Resource usage billed only as used

**Please Note:** Write your answers in a separate document and *save it locally* before submitting the answers here. Resubmitting the form (e.g. by answering one question at the time) clears all previous answers.

Questions:

1. Briefly discuss each of these principles. Are they necessary? What do they mean for a cloud service? What do they mean for a service using the cloud?
2. Is this a complete set? Are there other [fundamental] principles that you expect for cloud computing?
3. Elastic Scaling is one of the key principles. Please reason about the business implications of this.
   1. Give examples of applications where a quick (within minutes) elastic scaling might be necessary.
   2. Give examples of applications where a slower scaling would be more relevant.
   3. If your application grows at a steady state, what does this mean for your scaling needs? Is "going to the cloud" still the best option?

# PA2542 – Assignment What is Cloud Computing?

**Pooled Computing Resources** states that cloud resources shall be available to any subscribing user. This means that the resources are not dedicated for a specific system or application. This also means that different solutions share the cloud resources. For the cloud providing service, the pooling is not technically necessary, but without pooled computing resources, and virtualization discussed in the next section, there is no business case for the cloud provider. For the service using the cloud this means the initial costs for launching the service are low, because of the shift from capital expenses (CAPEX) to operational expenses (OPEX).

**Virtualization of compute resources** is the cornerstone of the cloud. With virtualization big scaling became possible. For the service provider the virtualization means that the expensive hardware can be better utilized, and the CAPEX is distributed over the virtualized servers. The service using the virtualized server perceives the virtualized server as ordinary server.

**Elasticity scaling up and down** is made possible by the ease to set up new cloud resources as needed by provisioning instead of manual efforts. And with the same ease the resources can be terminated when the peaks are over. This off course means increased costs for the scaled need, but the service user does not need to set up permanent resources to cope with estimated peak usage.

The elasticity is made possible by **automated creation of new virtual machines or deletion of existing ones.** The required resources and the order in which to they are to be provisioned are defined in scripts that are executed against the APIs of the cloud provider. In this way the set up will always be the same, and can be built, configured, provisioned, and moved, all without manual intervention.

These four principles together lead to the last principle: **Resource usage billed only as used**. I think this is one of the major drivers for the move towards the cloud. For a low cost, Startups can easily setup the needed resources on the cloud to develop new solutions, and when ready for launch these resources can be moved to a production environment and scaled to meet the increased need. No need to purchase, host and maintain an adequate server farm with high CAPEX, and significant OPEX. Also established enterprises benefit from the service that the cloud provides. As the cloud provider has taken on the responsibility for the maintenance of the cloud, the personnel that previously was needed for maintenance of the infrastructure now can be shifted to staff actually developing the product.

As the cloud technology matures, and with the increasingly finer granulation of the cloud resources offered by the cloud providers, the responsibility for the setup, the maintenance and security of the resources has moved from the customer to the service providers. Initially, a whole virtual server was provisioned, like AWS EC2, where the customer had to make sure that the correct server image was used, the server updated, and upgraded as needed. On this virtual box then the required resource, a database server, a web server or an application server was installed. With the entrance of serverless functions, like Lambda or Azure functions, messaging ques, and databases as a service, the cloud provider now takes on this responsibility. I would argue that this a further principle that will be more and more fundamental.

Elastic Scaling is one of the main principles of the cloud. As previously outlined, this enables the service user to adjust the required resources to the momentary situation. I think many people remember 9/11 2001 when the two planes crashed into the Twin Towers on Manhattan. Millions of people in Sweden who wanted to be up to date tried to follow the news on Aftonbladet.se, one of the most popular news sites in Sweden at that time. Due to the many requests, the web servers couldn’t cope anymore. The suddenly increased pressure on the servers turned into a natural DDOS attack. Another example where elastic scaling is beneficial is when ticket selling sites releases tickets to very popular artists. If the cloud resources couldn’t quickly scale up, the sites would be clogged down from the tsunami of requests.

On the other side, an arranged DDOS attack might be a situation where it might be better not to scale up, as the costs to ride out the DDOS could be immense. If the provided functionality is not of vital importance, it could be wise to shut down the service instead of scaling up.

With slow but steadily increase of needed cloud resources, it is possible to anticipate the costs and by that calculate whether going the cloud way is the best strategic. Here factors like CAPEX, OPEX needs to be taken in account, but also other factors need to be considered. Factors like whether there are there regulatory requirements that advocates an in-hose infrastructure, or how the possibility is to employ personal needed to support and maintain the locally hosted solution.

But with the increasing acceptance of cloud resources, and the cloud providers getting more and more transparent with their compliance to internationally established standards, it’s getting harder to justify hosting your own cloud resources.