I am not sure whether the key for this assignment is to make a summary or do questions answering. To make it clear, I would just make the writing into two sections.

Section 1 - Paper Summary

The paper gives a proactive definition of cloud computing, which clarifies the two layers of cloud hardware providers and SaaS service providers. Based on the conceptual framework, it discusses the benefits of cloud computing, and why it is a good time (not today, but in 2009) for the industry to develop cloud computing technics and products. The paper also proposes the top 10 obstacles to adopting cloud computing and the potential solution for each one.

Section 2 - Question Answering

Question 1

When talking about computing as pure utilities, this paper doesn't explicitly say "computing can become utilities". Instead, it summarizes the current (2009) use cases for cloud computing and implies the similarity between the two to some extent:

1. It highlights the ability to dynamically add/remove compute capacity, allowing usage to scale up and down based on needs. This does match the on-demand nature of utilities.
2. Pay-as-you-go pricing is also discussed, aligning with utility billing models.
3. Near infinite capacity and no upfront costs are convincing parallels to utilities.

Beyond that, this paper also illustrates several points that cloud computing can do better so that becomes more similar to "real utilities":

1. It notes there can still be multi-hour lead times to acquire additional resources on clouds like EC2, unlike utilities where additional capacity is instant.
2. Clouds do still experience outages and limitations on capacity, unlike the always-on nature of water. Stability and security are becoming more and more important.

Question2

1. An example that the analogy still applies:

Nowadays AWS is the choice for most business users. AWS Lambda is a Serverless computing service that allows a software engineer to just write and upload code without managing any servers. The cloud provider handles all the infrastructure, scaling, availability, etc. This is analogous to utilities where you don't manage the power plant or grid, just like using water from the tape.

Reference: https://aws.amazon.com/lambda/

2. An example that the analogy doesn't apply:

With IaaS like EC2, the software engineer is still responsible for managing the guest OS, security patches, network config, etc inside the VMs. The shared responsibility model means the utility analogy breaks down as the engineer has to handle many operational aspects utility users don't.

Reference: https://aws.amazon.com/compliance/shared-responsibility-model/