I think an Availability Zone would cost more than an Availability Set. An Availability Zone means deploying either a redundant VM or a service across multiple datacenters within a region, whereas an Availability Set means deploying either a redundant VM or a service on different sets of hardware (probably different racks?) within a single datacenter (within a region). Deploying services across multiple datacenters will be more costly than within a single datacenter, because the physical equipment (power, cooling, backup power generators) and infrastructure (metro high speed network connectivity) to mirror data is generally more expensive between datacenters than within a single one. Availability Zones provide 4 9's (99.99%) uptime, while Availability Sets provide 3 and a half 9's (99.95%) uptime.

Let's consider 2 different types of department store.

Let's assume the 1st department store is a small "mom and pop" department store that specializes in selling bowling balls to both bowlers and nearby bowling alleys. Because bowling balls are heavy and aren't conducive to shipping/e-commerce, all of department store's inventory is sold in their brick and mortar store. They do have a point of sale system which also keeps track of their inventory. The point of sale system has logic built into it where transactions are queued up in the event the network or application goes down. Fortunately, Mom and Pop are also quite organized and have a system to organize the different weights, colors, and manufacturers of their bowling balls, such that they can find their inventory quickly even when their real time inventory system is down. Mom and Pop "Bowl" have decided that because they do not require a high degree of uptime for their application and because they want to reduce costs as much as possible, they decide against any redundancy. In this case, the cost of availability sets and availability zones do not benefit the business bottom line to warrant the cost.

Let's assume the 2nd department store is a large clothing retailer that has both brick and mortar stores and an e-commerce site. One of the most critical parts of the business is a heavily utilized Order Management and Inventory database that attempts to keep inventory in realtime for both the stores and the e-commerce site. Every transaction in store and on the e-commerce eventually must be reflected in this single database, and while inventory of the company's sku's would ideally be reflected in realtime, the company has determined that some latency is acceptable. Moreover the company has employed a similar strategy as "Mom and Pop Bowl" when it comes to handling in-store purchases—transactions will be queued if the POS goes down. After doing some behavioral research, the company has found that most of their online customers come back to complete their orders if for some reason the site is experiencing technical issues when they checkout.

The company's management has also been heard saying "None of our customer's have ever died because they couldn't place an order for a new pair of pants." For that reason, the company has decided that 3 9's (99.9%) uptime is a suitable balance between costs and uptime. The IT staff of the company decide that using Availability Sets makes sense for the Order Management and Inventory Database. The decide to have a clustered multi-master DB setup where one nodes can go down at anytime and clients will failover to the other node. With this type of DB and the 99.95% SLA afforded by availability sets, the retailer should meet the business requirements.

Let's consider 2 different types of Banks.

Let's assume the 1st bank is a large regional bank that has several brick and mortar stores in New England and have roughly 10 million customers. They process bank transfers, direct deposit payments, have a network of 10,000 ATM's, and an e-commerce site that their clients use to check their balances and pay bills. The bank knows that trust with their clients in the banking industry is paramount and that downtime should be minimized whenever possible. That said, management understands that some issues are ultimately unavoidable and have coded many of their applications to operate with some built in graceful degradation. The development team has spent much time being able to catch exceptions and deliver user friendly messages to their customers during brief hiccups or outages. The banks management realizes that if their customer's cannot make payments to vendors or debtors, there can be lasting ramifications and ultimately unhappy customers, so they decide 99.99% uptime is desired. They employ their banking application across Availability Zones, so that if an entire datacenter within a region goes down, they have redundancy for their application in the other datacenter in the same region.

Let's assume the 2nd type of bank is the Federal Reserve (i.e. Fedwire https://en.wikipedia.org/wiki/Fedwire —the heart of the banking system). Every transaction that occurs between banks, the management of interest rates, and treasury exchanges go thru the federal reserve system. Problems within the Federal Reserve system can have catastrophic results. Mismanagement, downtime, errors, etc could have ramifications for the global economy, and potentially lead to war, and ultimately cost human life. Confidence in the heart of the banking industry and the dollar itself is of the highest importance and maximum redundancy should be considered. For that reason the Federal Reserve banking system has decided to use both Availability Sets (to insure redundancy within the datacenter) and Availability Zones (to insure redundancy within the region). Using both of these technologies should provide better than 4 9's (99.99%) uptime.
(^ note on above scenario. This is demonstrate understanding of availability sets/

zones. There would likely be some sensitive information stored within the Fed that might make it a poor choice for cloud altogether. In reality, something as important as the fed would also likely have redundancy across regions, and/or a DR site outside of the AZ region).