1. *(10 pts) Explain under-provisioning and over-provisioning in traditional datacenters. How does a cloud provide elastic computing?*

**Under-Provisioning**: This occurs when the capacity of the datacenter is not sufficient to meet the demand. In under-provisioned environments, there may be insufficient server capacity, not enough storage, or inadequate network bandwidth to handle peak loads effectively. This can lead to performance bottlenecks, slow response times, and system downtime, ultimately impacting the user experience and potentially leading to revenue loss.

**Over-Provisioning**: In contrast, over-provisioning happens when a datacenter has more resources than necessary. While this might sound like a safe approach, it leads to inefficiencies such as idle resources, wasted energy, and higher operational costs. Organizations often over-provision to ensure they can handle peak loads, which might only occur occasionally (e.g., during holiday sales in retail businesses).

The following are methods on how cloud services may provide elastic computing.

**Automatic Scaling**: Cloud services can automatically adjust the number of computing resources based on the application demand. This is typically handled by auto-scaling services that monitor your applications and automatically adjust capacity to maintain steady, predictable performance.

**Pay-as-You-Go Pricing**: Unlike traditional datacenters, where resources are paid for regardless of their use, cloud providers typically offer a pay-as-you-go model. This means that you only pay for the computing resources you actually use. This pricing model further encourages optimizing the amount of provisioned resources.

**Resource Management Tools**: Cloud providers offer sophisticated management tools that help organizations monitor their resource usage, predict demand, and trigger scaling actions. These tools use metrics and analytics to provide visibility and control over resource allocation.

**Rapid Provisioning**: Cloud environments enable rapid provisioning and deprovisioning of resources. This allows organizations to deploy additional resources within minutes to handle unexpected increases in load and then scale down as soon as the demand decreases, thus optimizing both cost and performance.

Examples from Major Cloud Providers

**AWS**: Amazon Web Services offers services like AWS Auto Scaling, Elastic Load Balancing, and AWS Elastic Beanstalk, which handle the scaling of resources across multiple services like Amazon EC2 instances, databases, and storage.

**Google Cloud**: Google Cloud Platform provides similar functionalities with Google Compute Engine autoscalers and custom machine types that can be tailored to fluctuating demands.

**Azure**: Microsoft Azure features Azure Autoscale and virtual machine scale sets to adjust resources dynamically in response to workload changes.

1. *(10 pts) Compare Amazon AWS, Google cloud, and Microsoft Azure cloud in terms of services they provide.*

**AWS**: Offers Amazon EC2 (Elastic Compute Cloud) for scalable computing capacity, AWS Lambda for serverless computing, and Amazon Lightsail for simpler applications.

**Google Cloud**: Provides Google Compute Engine for VMs, Google App Engine for PaaS, and Google Cloud Functions for serverless computing.

**Azure**: Features Azure Virtual Machines, Azure Functions for serverless computing, and Azure App Service for web and mobile apps.

1. *(10 pts) How do Amazon AWS and Google Cloud achieve auto-scaling?*

There are 3 types of auto scaling policies available in AWS:

* First one is a target tracking scaling policy. Here scaling is defined by the target value of a metric (CPU, request count). The autoscaling will try to maintain the metric’s value as close to the target value by increasing or decreasing the instance count.
* The second one is a step-based auto-scaling policy. It gives the ability to add the number of instances as per the level of alarm breach. For example, if the target CPU is 50% and the actual CPU has reached 60%, then you can choose to add 1 instance but if the CPU reached 80% then you can add 2 instances.
* The third one is a simple scaling policy. Here autoscaling will add the instances linearly until the target value is achieved.

For both step and simple scaling policy, we have to define separate scale out and scale in the policy. In case of target tracking scaling policy, we just have to give the target value and autoscaling will add or remove the instances as required. GCP only supports the target tracking scaling policy. So AWS has more scaling policy compared with the GCP.

1. *(10 pts) Explain REST protocol and the operations it provides.*

A REST API (also called a RESTful API or RESTful web API) is an [application programming interface (API)](https://www.ibm.com/topics/api) that conforms to the design principles of the *representational state transfer* (REST) architectural style. REST APIs provide a flexible, lightweight way to integrate applications and to connect components in [microservices](https://www.ibm.com/topics/microservices) architectures.

1. *(10 pts) How are buckets and containers used in Amazon and Azure clouds?*

In the AWS platform, cloud storage is primarily broken down into three services:

**Simple Storage Service (S3)**. Basic object storage that makes data available through an Internet accessible API.

**Elastic Block Storage (EBS)**. Block level storage intended for access by a single VM.

**Elastic File System (EFS)**. File storage meant for use as shared storage for up to thousands of EC2 instances.

In Azure Storage, subscription-bound [storage accounts](https://learn.microsoft.com/en-us/azure/storage/common/storage-quickstart-create-account) allow you to create and manage the following storage services:

[Blob storage](https://learn.microsoft.com/en-us/azure/storage/common/storage-quickstart-create-account) stores any type of text or binary data, such as a document, media file, or application installer. You can set Blob storage for private access or share contents publicly to the Internet. Blob storage serves the same purpose as both AWS S3 and EBS.

[Table storage](https://learn.microsoft.com/en-us/azure/cosmos-db/table-storage-how-to-use-nodejs) stores structured datasets. Table storage is a NoSQL key-attribute data store that allows for rapid development and fast access to large quantities of data. Similar to AWS' SimpleDB and DynamoDB services.

[Queue storage](https://learn.microsoft.com/en-us/azure/storage/queues/storage-quickstart-queues-nodejs?tabs=passwordless%2Croles-azure-portal%2Cenvironment-variable-windows%2Csign-in-azure-cli) provides messaging for workflow processing and for communication between components of cloud services.

[File storage](https://learn.microsoft.com/en-us/azure/storage/files/storage-java-how-to-use-file-storage) offers shared storage for legacy applications using the standard Server Message Block (SMB) protocol. File storage is used in a similar manner to EFS in the AWS platform.

1. *(10 pts) What’s the difference between web and worker roles in Azure cloud?*

There are two types of Azure Cloud Services roles. The only difference between the two is how your role is hosted on the VMs:

**Web role**: Automatically deploys and hosts your app through Internet Information Services (IIS).

**Worker role**: Doesn't use IIS, and runs your app standalone.

1. *(10 pts) What’s the difference between full virtualization and para-virtualization?*

The key difference between full virtualization and paravirtualization is that in full virtualization Virtual Machines (VM) do not share any resources and are completely isolated while paravirtualization allows for sharing of communication and resources with an operating system.

1. *(30 points) Using the EC2 you created in HW2, write a program to test whether a given year is a leap year or not and run this program on your EC2. Display your results on the webserver you created in HW2 in the following format for years 1996, 2000, 2006, 2023, and 2024:*

*“Year xxxx is a leap year” or*

*“Year xxxx is not a leap year.”*

*Submit your code, a screenshot to show that you run your code on your EC2 (not your local machine), and a screenshot of the display of the results on your web server.*



