**TASK 10 – Launch an instance using Yellow Circle Cloud.**

**AIM :** To Create and launch Windows Server virtual machine using Yellow Circle Cloud Platform.

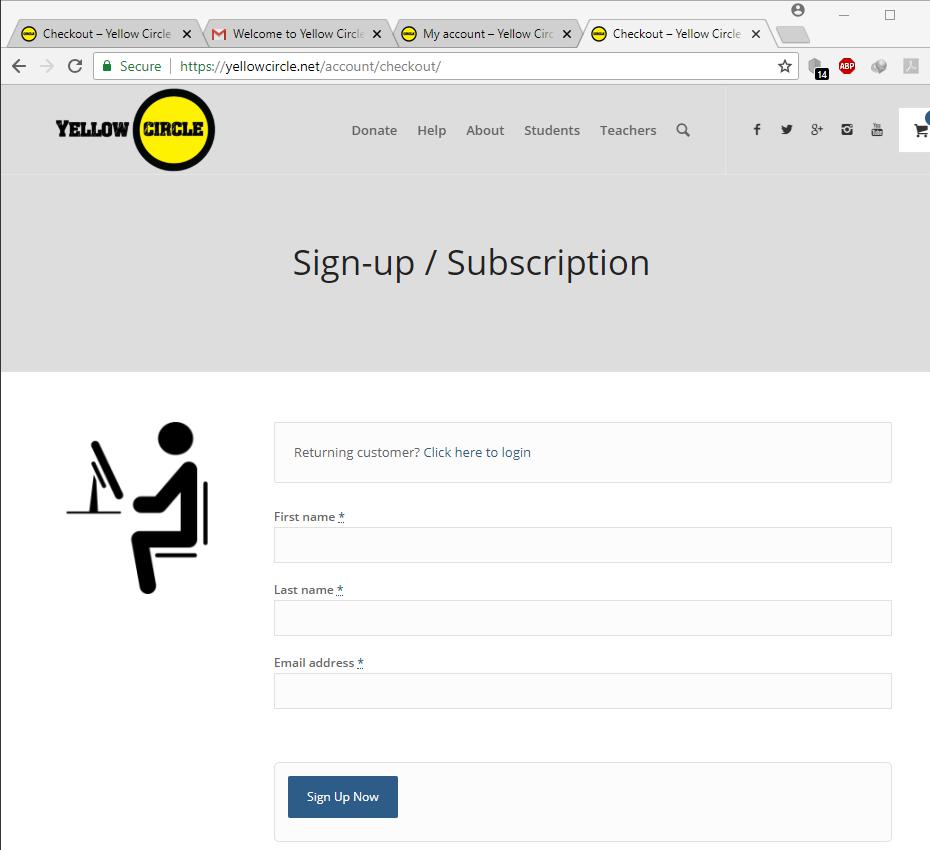
**DESCRIPTION :**

Yellow Circle provides a virtual information technology platform allowing students to learn about the unique capabilities of cloud computing and cloud infrastructure. Yellow Circle’s ultimate mission is to provide a platform for building the foundation for our future IT workforce to produce sustainable, equitable IT solutions to a variety of global technology challenges.

In this experiment we will learn how to launch an instance and run using yellow circle.

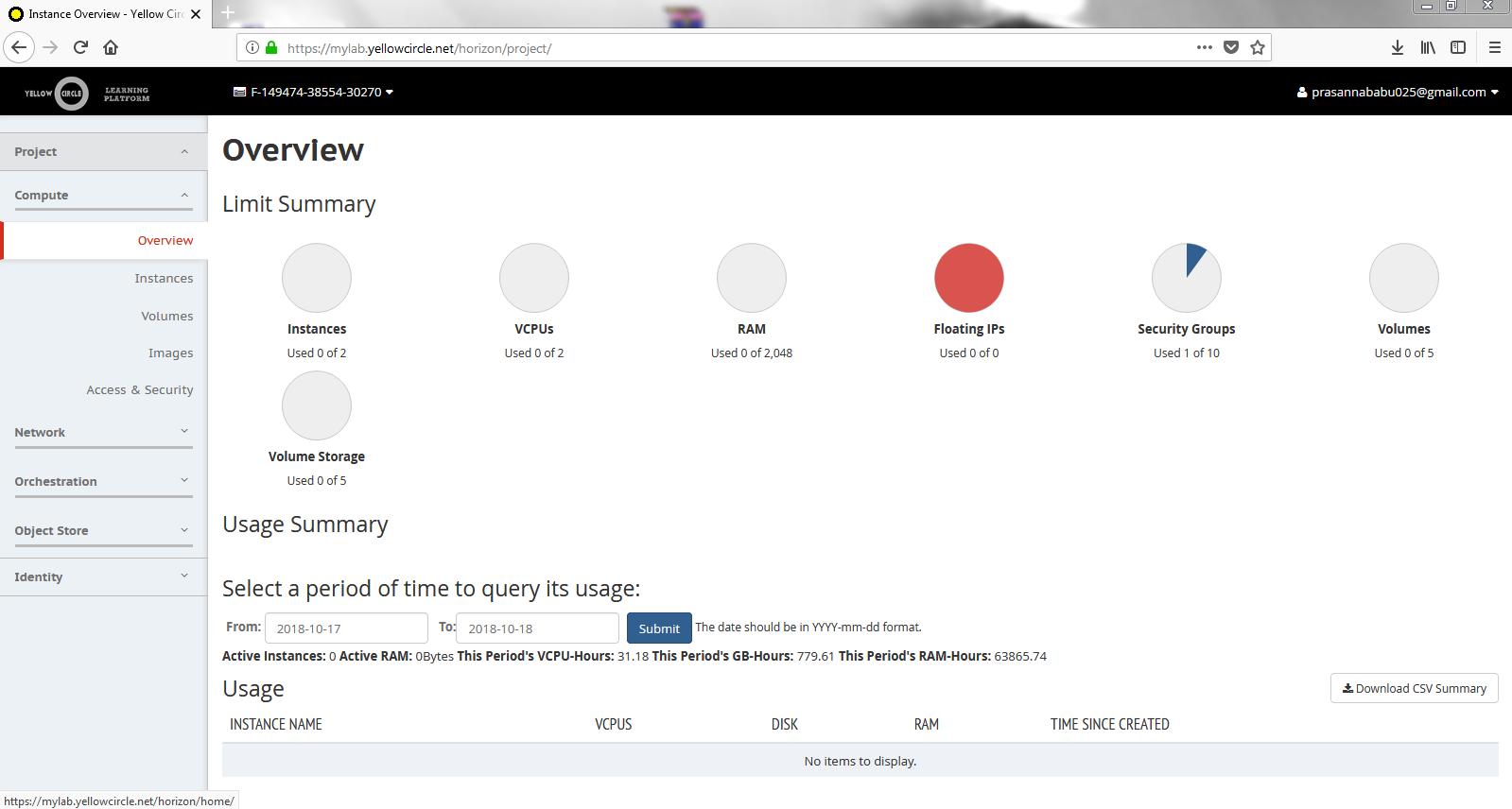
**PROCEDURE :**

1. First go to [www.yellowcircle.net](http://www.yellowcircle.net) and signup for a free student account.

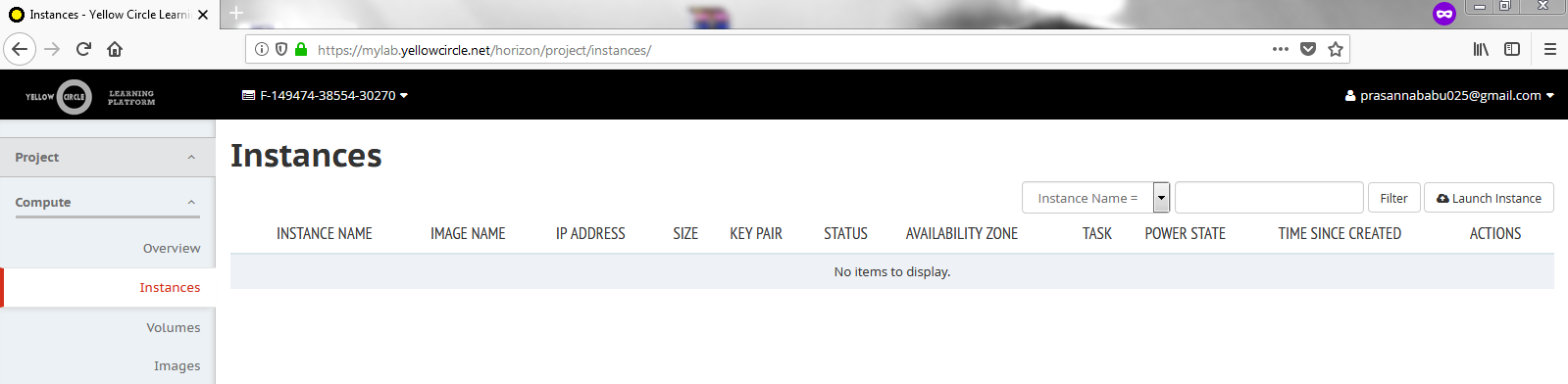


After signup you will get a mail with the login id and a password to login to your yellow circle account.

After logging in the dash board looks as follows :

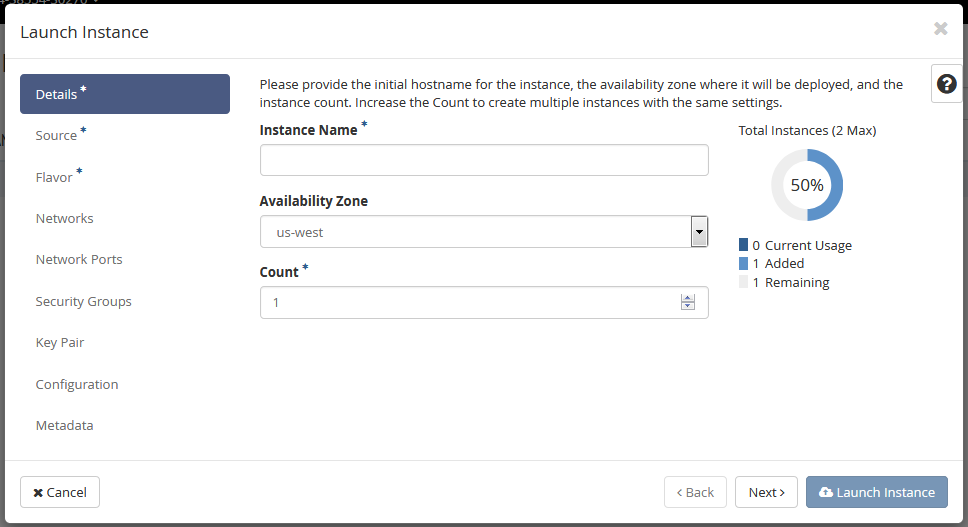


2. Click on the Instances from the Compute menu located at the left side .

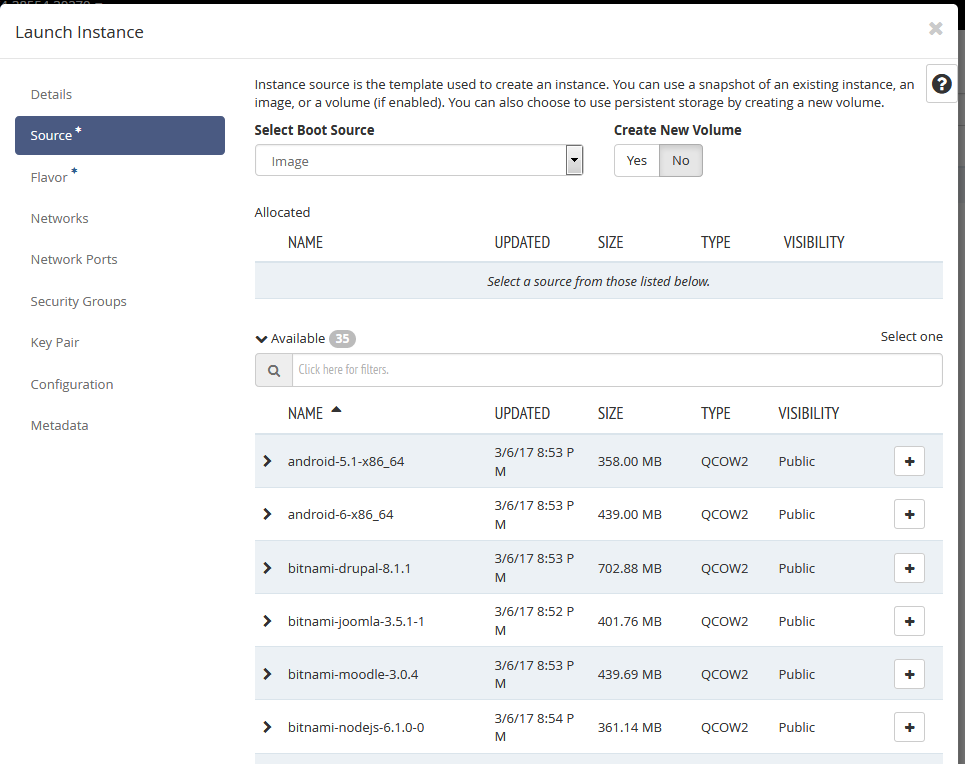


3. Now Click on the Launch instance

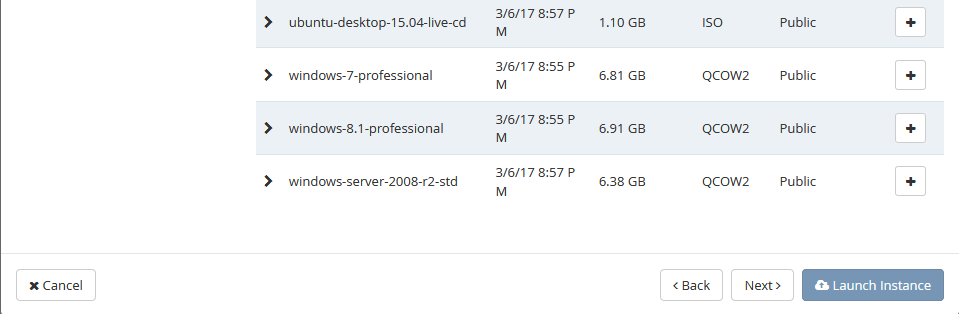
Enter the name of the Instance

This is how the window looks like .

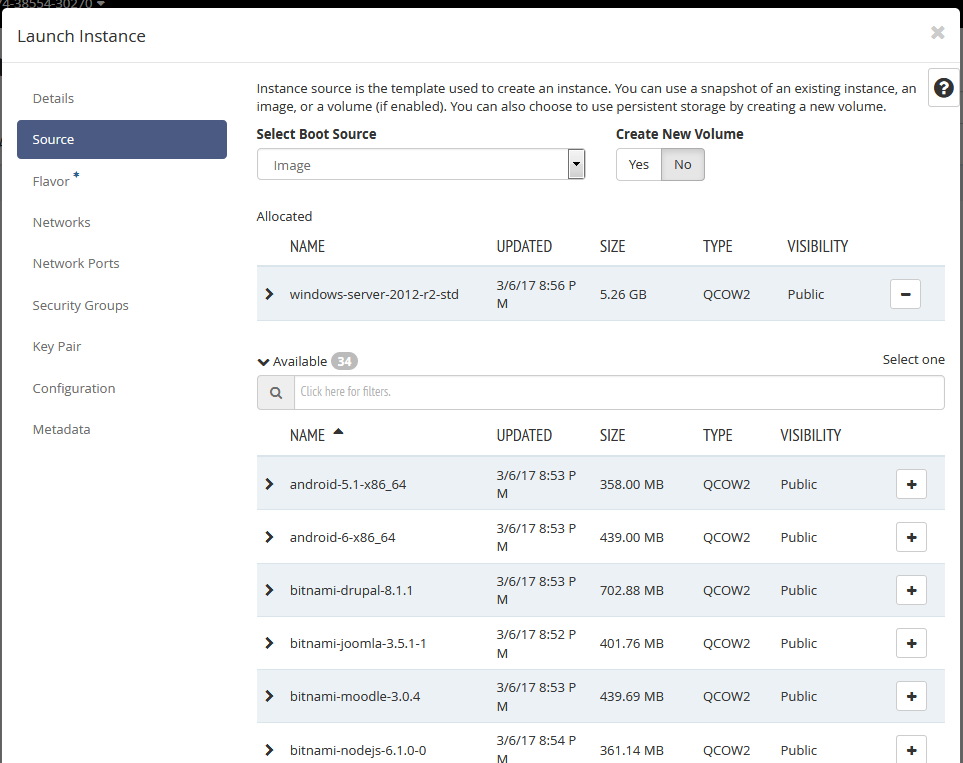
Click Next , you will be prompted to another window which shows the available instance sources.



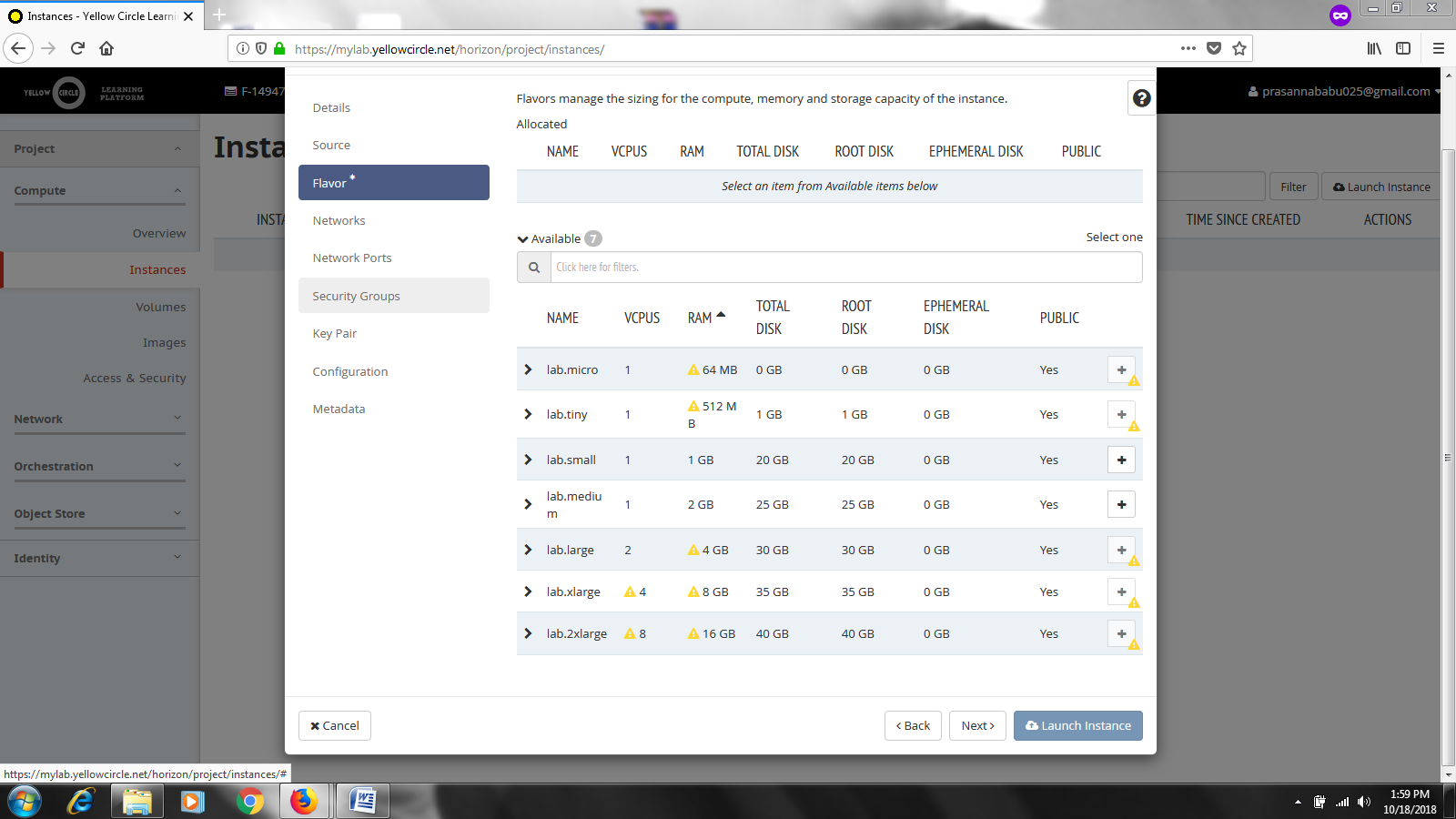
4.Now select the windows-server-2008-r2-std by clicking the + button and click Next .



The window looks like this after selecting an instance.

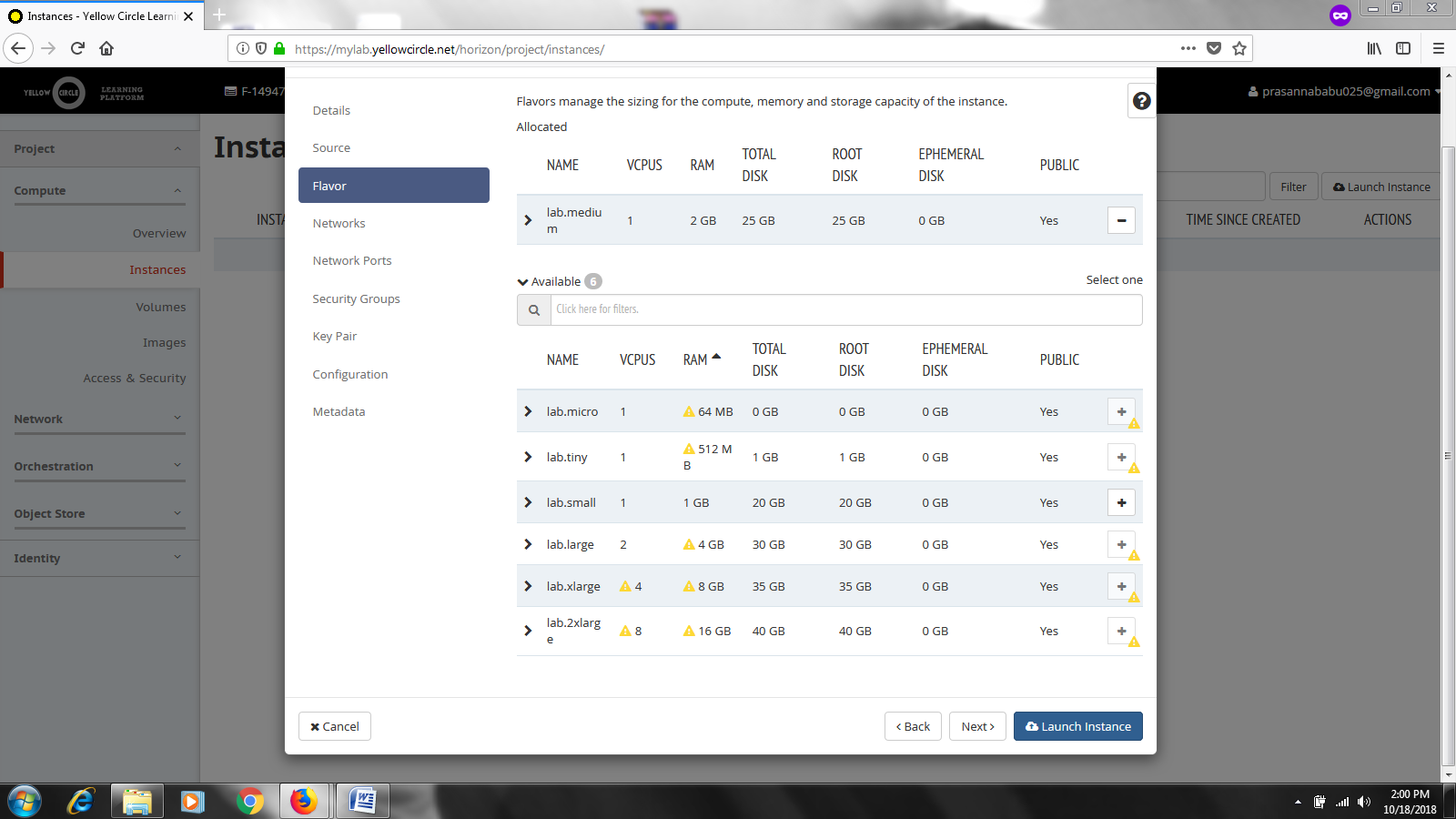


5. Now select and click Flavor at the left menu , select lab.medium and click next.

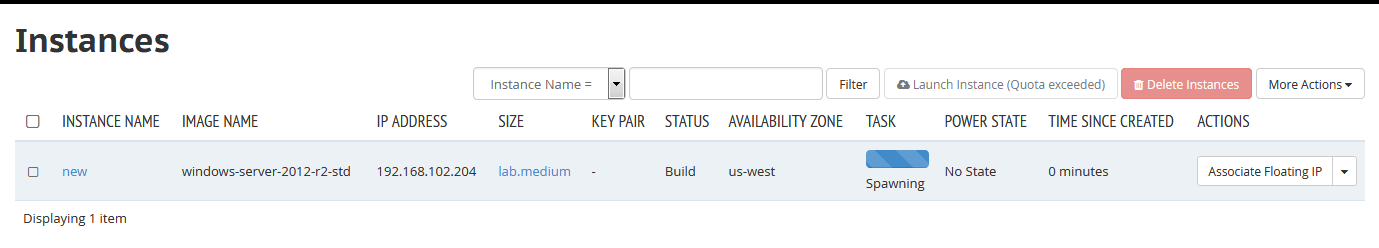


The window looks as follows after selecting the lab.medium flavor.

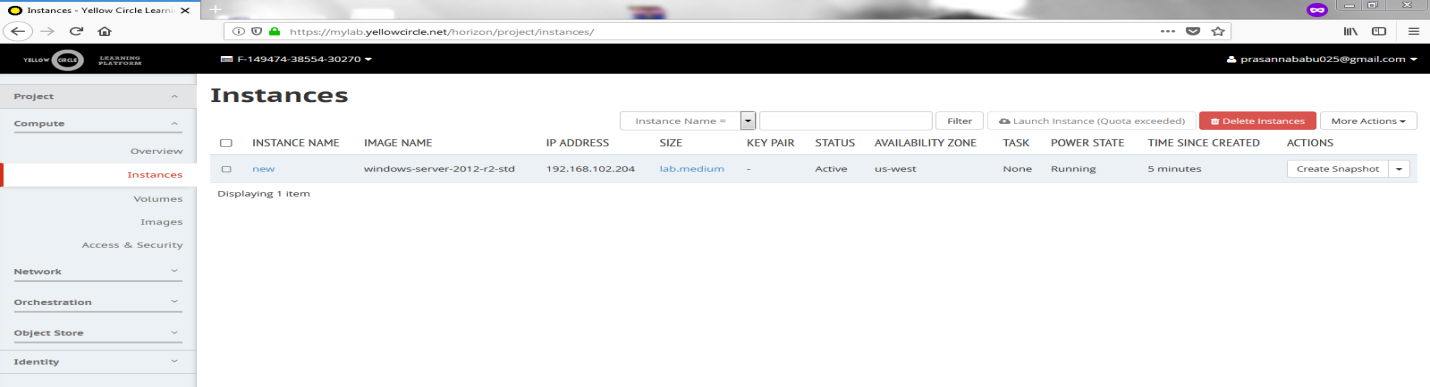
6.Click Launch Instance.



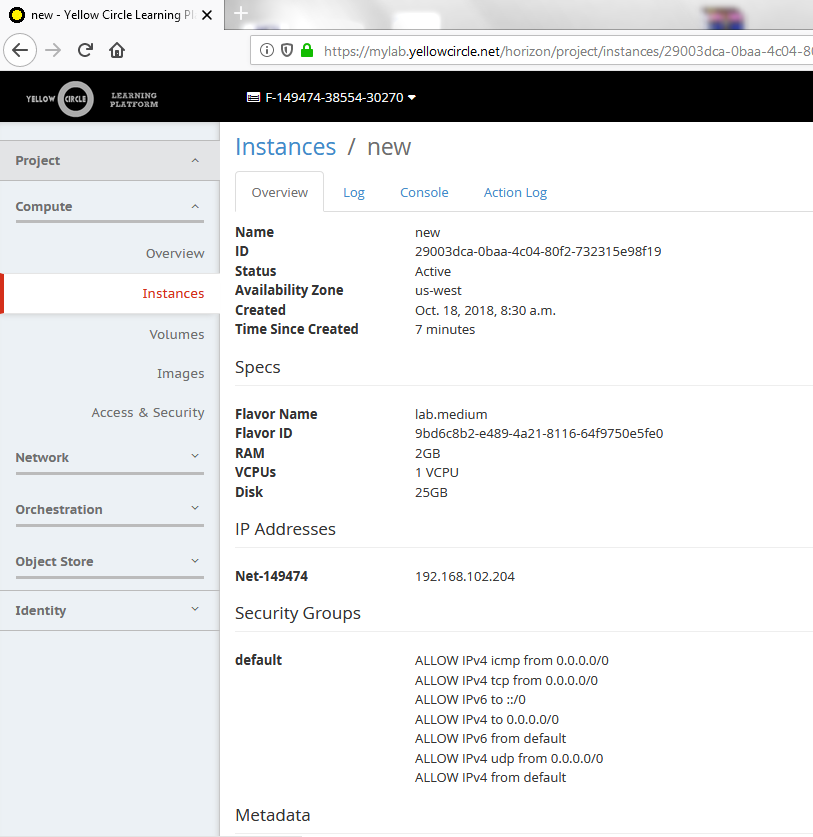
Now Your instance will begin to spawn as shown below.



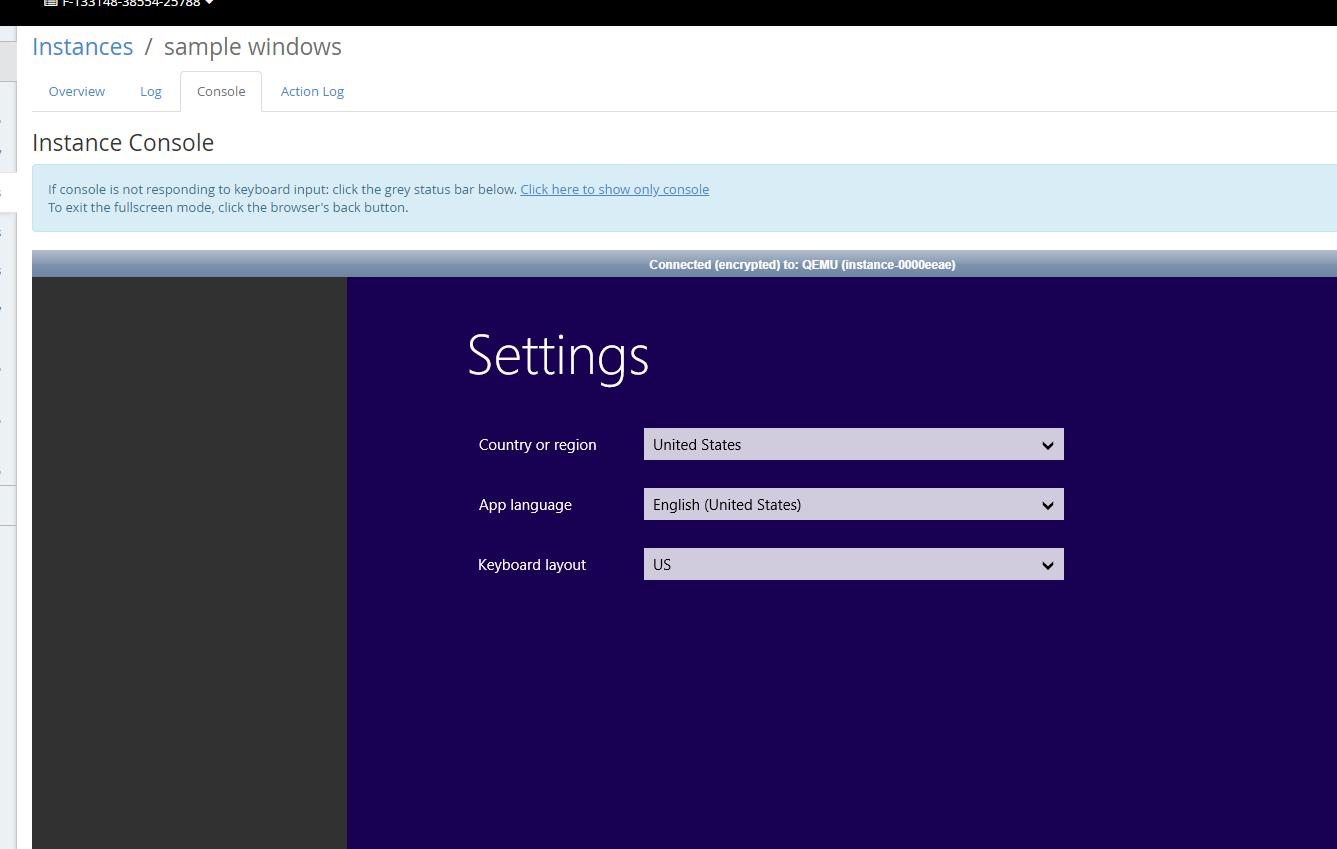
Within a few moments your instance will be ready. It will stop being orange, the status will be Active and the Power state will be in Running mode



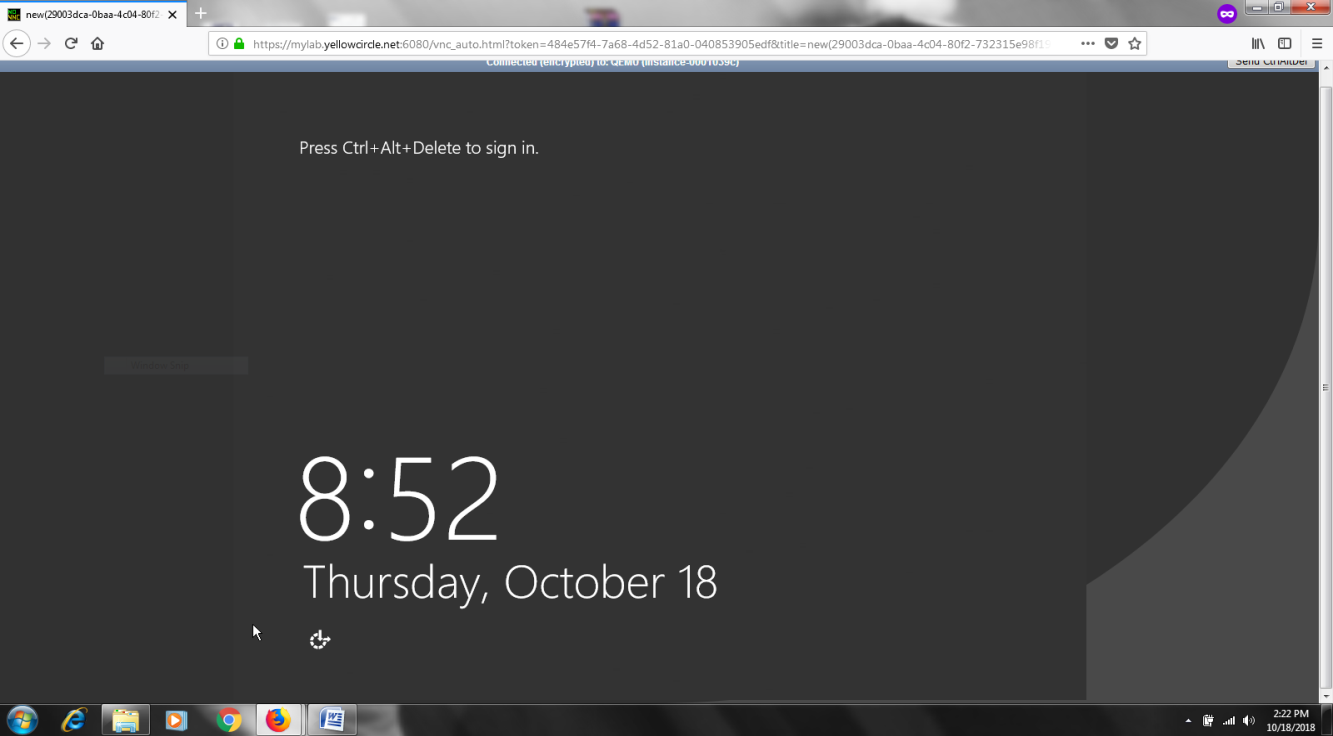
7.Click on the Name of your instance and select Console to start the windows machine



We get the following screen, Click next and set the password to run the windows machine.



Thus Windows machine is ready .



**RESULT :** Hence the creation and launching of an instance (Windows server ) using Yellow circle completed successfully.

**TASK 11 – Case Studies**

**AIM:** To study about Hadoop , Aneka

**Hadoop**

Hadoop is an open-source software framework for storing data and running applications on clusters of commodity hardware. It provides massive storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs.

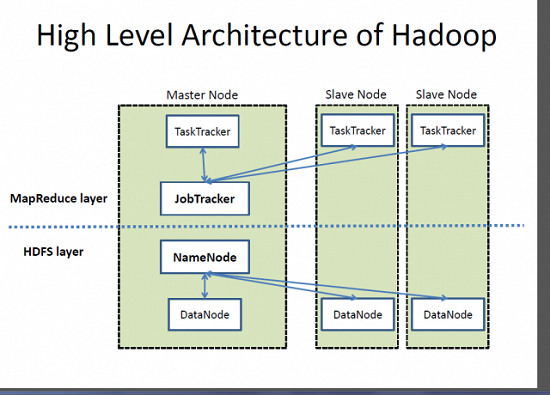
**Hadoop Architecture**

Apache Hadoop offers a scalable, flexible and reliable distributed computing big data framework for a cluster of systems with storage capacity and local computing power by leveraging commodity hardware. Hadoop follows a Master Slave architecture for the transformation and analysis of large datasets using [Hadoop MapReduce](https://www.dezyre.com/hadoop-course/mapreduce" \t "_blank" \o "What is Hadoop MapReduce?) paradigm. The 3 important hadoop components that play a vital role in the Hadoop architecture are –

* Hadoop Distributed File System (HDFS) – Patterned after the UNIX file system
* Hadoop MapReduce
* Yet Another Resource Negotiator (YARN)

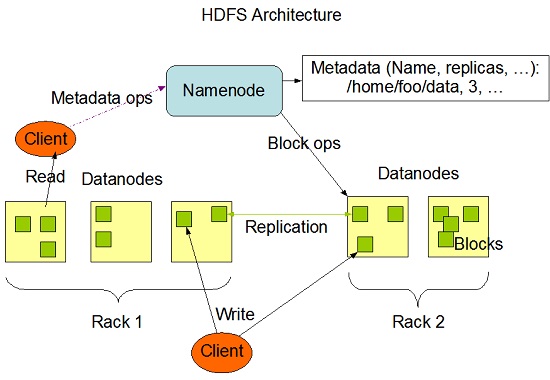
[Hadoop skillset](https://www.dezyre.com/hadoop-course/hadoop-training-in-california) requires thoughtful knowledge of every layer in the hadoop stack right from understanding about the various components in the hadoop architecture, designing a hadoop cluster, performance tuning it and setting up the top chain responsible for data processing.

Hadoop follows a master slave architecture design for data storage and distributed data processing using [HDFS](https://www.dezyre.com/hadoop-course/hdfs) and MapReduce respectively. The master node for data storage is hadoop HDFS is the NameNode and the master node for parallel processing of data using Hadoop MapReduce is the Job Tracker. The slave nodes in the hadoop architecture are the other machines in the Hadoop cluster which store data and perform complex computations. Every slave node has a Task Tracker daemon and a DataNode that synchronizes the processes with the Job Tracker and NameNode respectively. In Hadoop architectural implementation the master or slave systems can be setup in the cloud or on-premise.



### **Role of Distributed Storage – HDFS in Hadoop Application Architecture Implementation**

A file on HDFS is split into multiple bocks and each is replicated within the Hadoop cluster. A block on HDFS is a blob of data within the underlying file system with a default size of 64MB.The size of a block can be extended up to 256 MB based on the requirements.



Hadoop Distributed File System (HDFS) stores the application data and file system metadata separately on dedicated servers. NameNode and DataNode are the two critical components of the Hadoop HDFS architecture. Application data is stored on servers referred to as DataNodes and file system metadata is stored on servers referred to as NameNode. HDFS replicates the file content on multiple DataNodes based on the replication factor to ensure reliability of data. The NameNode and DataNode communicate with each other using TCP based protocols. For the Hadoop architecture to be performance efficient, HDFS must satisfy certain pre-requisites –

* All the hard drives should have a high throughput.
* Good network speed to manage intermediate data transfer and block replications.

#### **NameNode**

All the files and directories in the HDFS namespace are represented on the NameNode by Inodes that contain various attributes like permissions, modification timestamp, disk space quota, namespace quota and access times. NameNode maps the entire file system structure into memory. Two files fsimage and edits are used for persistence during restarts.

* Fsimage file contains the Inodes and the list of blocks which define the metadata.It has a complete snapshot of the file systems metadata at any given point of time.
* The edits file contains any modifications that have been performed on the content of the fsimage file.Incremental changes like renaming or appending data to the file are stored in the edit log to ensure durability instead of creating a new fsimage snapshot everytime the namespace is being altered.

When the NameNode starts, fsimage file is loaded and then the contents of the edits file are applied to recover the latest state of the file system. The only problem with this is that over the time the edits file grows and consumes all he disk space resulting in slowing down the restart process. If the hadoop cluster has not been restarted for months together then there will be a huge downtime as the size of the edits file will be increase. This is when Secondary NameNode comes to the rescue. Secondary NameNode gets the fsimage and edits log from the primary NameNode at regular intervals and loads both the fsimage and edit logs file to the main memory by applying each operation from edits log file to fsimage. Secondary NameNode copies the new fsimage file to the primary NameNode and also will update the modified time of the fsimage file to fstime file to track when then fsimage file has been updated.

#### **DataNode**

DataNode manages the state of an HDFS node and interacts with the blocks .A DataNode can perform CPU intensive jobs like semantic and language analysis, statistics and machine learning tasks, and I/O intensive jobs like clustering, data import, data export, search, decompression, and indexing. A DataNode needs lot of I/O for data processing and transfer.

On startup every DataNode connects to the NameNode and performs a handshake to verify the namespace ID and the software version of the DataNode. If either of them does not match then the DataNode shuts down automatically. A DataNode verifies the block replicas in its ownership by sending a block report to the NameNode. As soon as the DataNode registers, the first block report is sent. DataNode sends heartbeat to the NameNode every 3 seconds to confirm that the DataNode is operating and the block replicas it hosts are available.

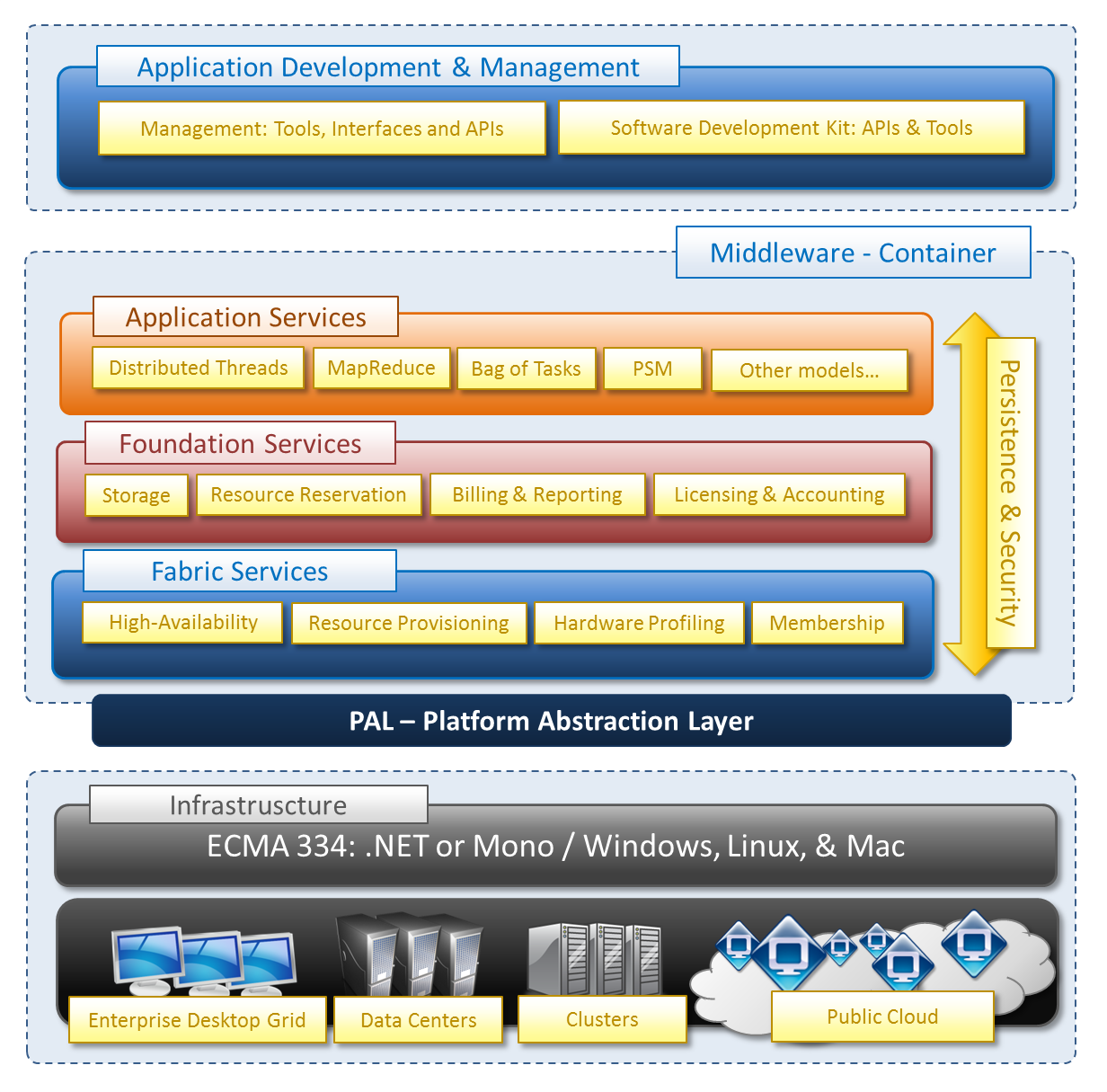
**Aneka**

Aneka is a platform and a framework for developing distributed applications on the Cloud. It harnesses the spare CPU cycles of a heterogeneous network of desktop PCs and servers or datacenters on demand. Aneka provides developers with a rich set of APIs for transparently exploiting such resources and expressing the business logic of applications by using the preferred programming abstractions. System administrators can leverage on a collection of tools to monitor and control the deployed infrastructure. This can be a public cloud available to anyone through the Internet, or a private cloud constituted by a set of nodes with restricted access.

The Aneka based computing cloud is a collection of physical and virtualized resources connected through a network, which are either the Internet or a private intranet. Each of these resources hosts an instance of the Aneka Container representing the runtime environment where the distributed applications are executed. The container provides the basic management features of the single node and leverages all the other operations on the services that it is hosting. The services are broken up into fabric, foundation, and execution services. Fabric services directly interact with the node through the Platform Abstraction Layer (PAL) and perform hardware profiling and dynamic resource provisioning. Foundation services identify the core system of the Aneka middleware, providing a set of basic features to enable Aneka containers to perform specialized and specific sets of tasks. Execution services directly deal with the scheduling and execution of applications in the Cloud.

**Aneka Architecture**

One of the key features of Aneka is the ability of providing different ways for expressing distributed applications by offering different programming models; execution services are mostly concerned with providing the middleware with an implementation for these models. Additional services such as persistence and security are transversal to the entire stack of services that are hosted by the Container. At the application level, a set of different components and tools are provided to: 1) simplify the development of applications (SDK); 2) porting existing applications to the Cloud; and 3) monitoring and managing the Aneka Cloud.



A common deployment of Aneka is presented at the side. An Aneka based Cloud is constituted by a set of interconnected resources that are dynamically modified according to the user needs by using resource virtualization or by harnessing the spare CPU cycles of desktop machines. If the deployment identifies a private Cloud all the resources are in house, for example within the enterprise. This deployment is extended by adding publicly available resources on demand or by interacting with other Aneka public clouds providing computing resources connected over the Internet.