**Case Study: E-Commerce Data Analysis Using Hadoop**

# updating 2500 BI dashboards daily for data-driven decisions

Project By,

Akshay Kolhapure

[akshaykolhapure995c@gmail.com](mailto:akshaykolhapure995c@gmail.com)

(+91)-8600 669 246

* **Abstract**

*E-commerce*is a transaction of buying or selling online. It draws on technologies such as mobile commerce, electronic funds transfer, supply chain management, and much more. The e-commerce field is increasing rapidly across the world. The e-commerce business in India will be worth 38 billion dollars by 2016 and it is estimated to reach 159 billion dollars by 2020.  
The e-commerce firms are growing rapidly all over the world with millions of transactions made every day. So, one needs to analyze that data and draw some useful insights from it.

Here, I bring to you a business use case of an e-commerce company which wants to analyze their transactions and draw some useful insights out of it, which will be useful for their business development.

* **Challenge**

Industry (E-Commerce) executives seek fresh insights when it comes to customer needs. In any subscription model business, customer retention is vital as profitability can quickly erode with rapid customer turnover and high customer acquisition costs.

For Industry, data provides the key to not only understanding customer likes and dislikes but also predicting their preferences and needs. What recipes,ingredients, and meals might each household want to try, and when would they most likely buy those meals?

However, as its subscriber base grew, Industry staff needed a new data warehouse that could support more data and more business users, and that could more accurately predict customer behavior.

Industry’s legacy SQL database was hard to scale and difficult to bring in new data and users. It also had slow performance. It would take hours to figure out how many boxes were shipped, and as we hired more analysts and data scientists, it became quite difficult even to maintain that level of performance.

* **Solution**

Industry is transforming analytics with a modern data warehouse,hadoop components and cloud. The platform, which runs on [cloud](https://www.cloudera.com/partners/solutions/amazon-web-services.html) , brings in and analyzes more than 15 terabytes (TB) of data to give staff a 360-degree view of how each customer interacts with the company for more personalized recommendations and more accurate forecasting.

More than 100 business users across industry, including fulfillment center managers, marketing staff, and product development teams, run thousands of queries daily on the platform to guide every level of operation. Additionally, users have easy access to insights in over 2,500 business intelligence (BI) dashboards which are updated daily.

Before implementation of hadoop architechure ,cloud and Cloudera Manager industry only could analyze about **five TB** of data

In less than a year, we’ve grown to analyze three times the amount of data, and we’re adding approximately one TB a month to the system. This includes unstructured and structured data on how customers interact with our products and service staff, and what feedback they provide in surveys

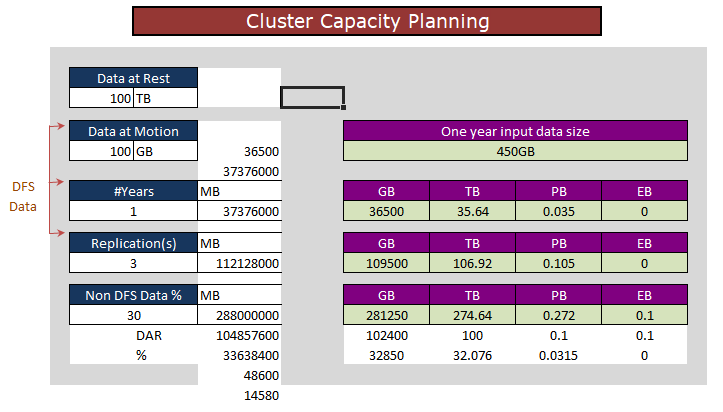
* **Cluster Capacity Planning**
* Volume of data for which cluster is being set : ***100 TB (100,000 GB)***
* Daily ingest : ***100 GB***
* The retention policy of the data : ***365 days (1 year)***
* Replication factor : ***3 replicas***
* The storage ratio : ***HDFS ->70% ,Non HDFS ->30%***
* The storage mechanism for the data : ***MYSQL , JSON***
* The kinds of workloads : ***For generalized analytics (CPU intensive)***

Planning For Worker Nodes :

With the above parameters in hand, we can plan for commodity machines required for the cluster. These might not be exactly what is required, but after installation, we can fine tune the environment by scaling up/down the cluster.The nodes that will be required depends on data to be stored/analyzed.

By default, the Hadoop ecosystem creates three replicas of data. So if we go with a default value of 3, we need storage of 100TB \*3=300 TB for storing data of one year.

We have a retention policy of one year.



**<Fig. (a) Total data calculation for 1 year retention policy>**

As per above calculation , we calculated total data (HDFS , non HDFS).

**Step by step explanation :**

During the period of POC, we found daily ingestion data as 1 GB (approx), data at rest is 100 TB.

Now , from above all steps of calculation we got **275 TB** including replication,storage ratio and number of period.

We have to manage **275 TB ( 281250 GB ) data** on hadoop for processing.

To find total number of nodes ,

Number Of Nodes ( Worker Nodes ) =(*Total size of data / Per node defined size )*

Here,

Daily Ingest Rate = 100 GB,

Replication Number = 3,

Number Of Days Of Year = 365 days (1 year),

Total Size Of Data = 275 TB

Per Node Defined Size (In TB) = 24 TB

Number Of Nodes ( Worker Nodes ) *=(275 / 24 TB)*

*=Round Off 12 nodes*

We have seen how we got 12 worker nodes, Now we have to distribute storage ,memory and CPUs among those nodes .

For that ,we will refer Cloudera’s rule of thumb.Cloudera recommend per node configuration for capacity planning.

|  |  |  |
| --- | --- | --- |
| RAM | CPU Cores | Storage |
| 32GB | 8 Cores | 2 X 2= 4TB |

Let’s find memory,CPUs,Stoarage to manage and process that much of data ,

Simply follow ,

On the 12 nodes we can distribute in same manner by dividing total data size and number of nodes ,

*275/12=22.91 ( Rnd off we will take 24 TB data per node)*

*Here we got per node data size ,* ***24 TB.***

Planning For Master Nodes :

Now let’s a take a step forward and plan for master nodes. For name nodes, we need to set up a failover name node, as well (also called a secondary name node). The secondary name node should be an exact or approximate replica of the primary name node.

Both name node servers should have highly reliable storage for their namespace storage and edit-log journaling. That’s why — contrary to the recommended JBOD for data nodes — RAID is recommended for name nodes.

Master servers should have at least four redundant storage volumes — some local and some networked — but each can be relatively small (typically 1TB).

It is easy to determine the memory needed for both name node and secondary name node. The memory needed by name node to manage the HDFS cluster metadata in memory and the memory needed for the OS must be added together. Typically, the memory needed by the secondary name node should be identical to the name node.

1. **Memory Sizing**

The amount of memory required for the master nodes depends on the number of file system objects (files and block replicas) to be created and tracked by the name node. We can do memory sizing as :

1. **64 GB of RAM supports approximately 100 million files**. So if you know the number of files to be processed by data nodes, use these parameters to get RAM size.
2. Or use this formula: **Memory amount = HDFS cluster management memory + NameNode memory + OS memory**.

OS memory 8 GB-16 GB, name node memory 8-32 GB, and HDFS cluster management memory 8-64 GB should be enough.

Name node memory and HDFS cluster management memory can be calculated based on the data nodes, and files to be processed. Use 1 and 2 to estimate these values.

1. **Processor**

Name nodes and their clients are very chatty. We, therefore, recommend providing 8 CPU cores for handling messaging traffic for the master nodes.

1. **Network**

Providing multiple network ports and 10 GB bandwidth to the switch is also acceptable (if the switch can handle it).

So finally,

We are using master node resources as follows :

1. *Namenode : 1 node*
2. ***Storage : 2 TB RAID Storage***
3. ***Memory : 128 GB RAM***
4. **CPU : 8 Core**

*b) Secondary Namenode : 1 Node*

1. ***Storage : 2 TB RAID Storage***
2. ***Memory : 128 GB RAM***
3. ***CPU : 8 Core***

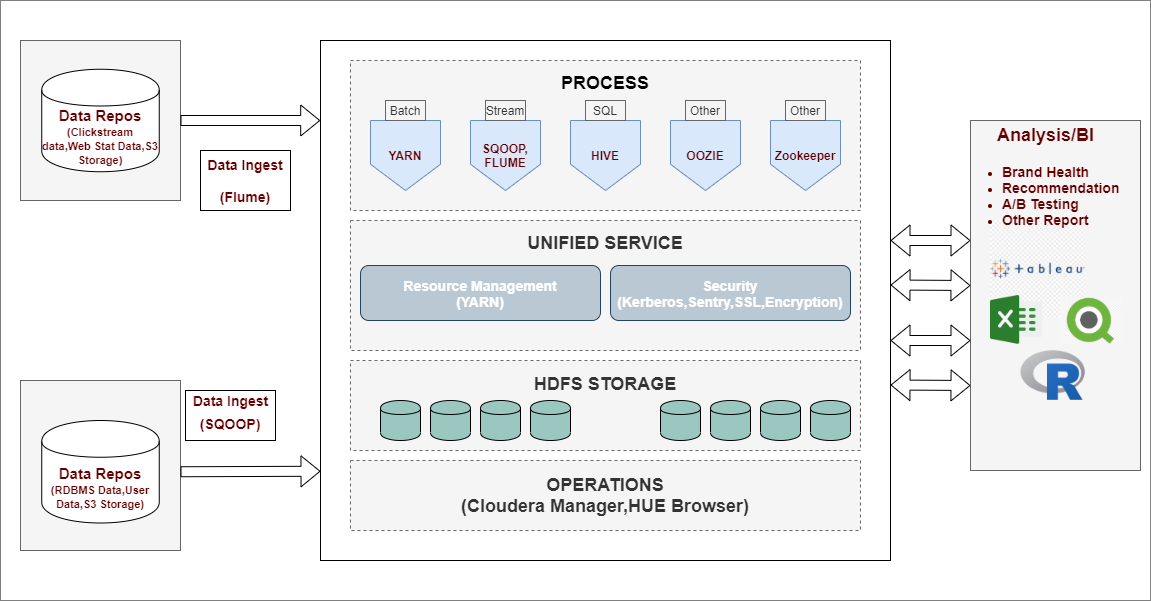
*c) Active Resource Manager : 1 Node*

1. ***Storage : 2 TB RAID Storage***
2. ***Memory : 128 GB RAM***
3. ***CPU : 8 Core***

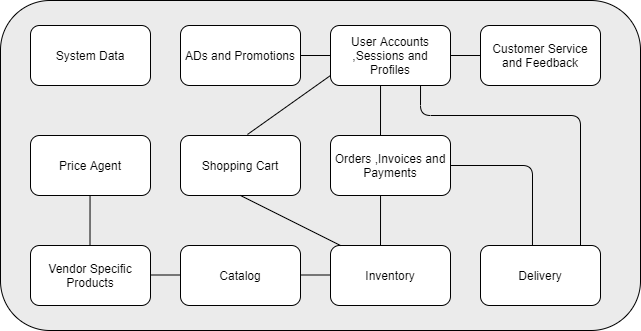
*d) Standby Resource Manager : 1 Node*

1. ***Storage : 2 TB RAID Storage***
2. ***Memory : 128 GB RAM***
3. ***CPU : 8 Core***

* **Hadoop Architecture**

****

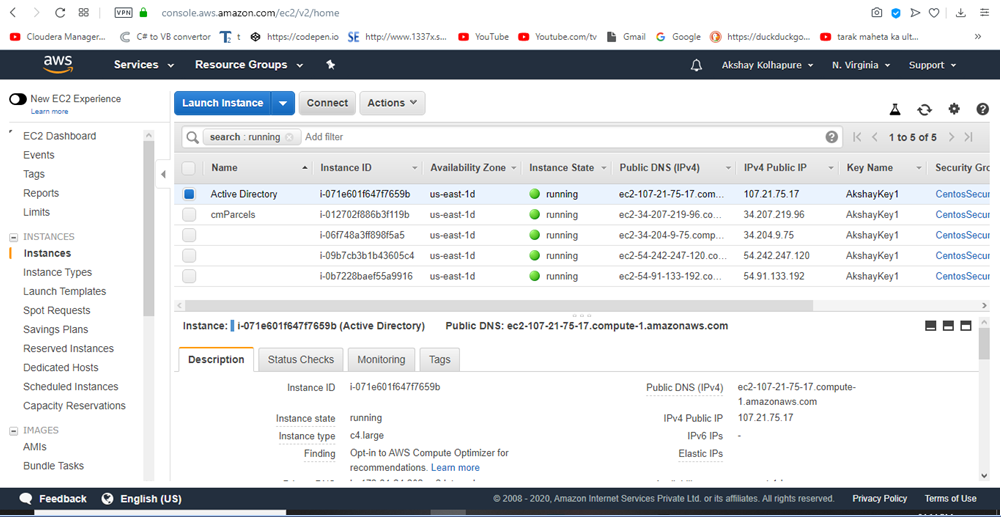
* **UML Diagram Of Database System**

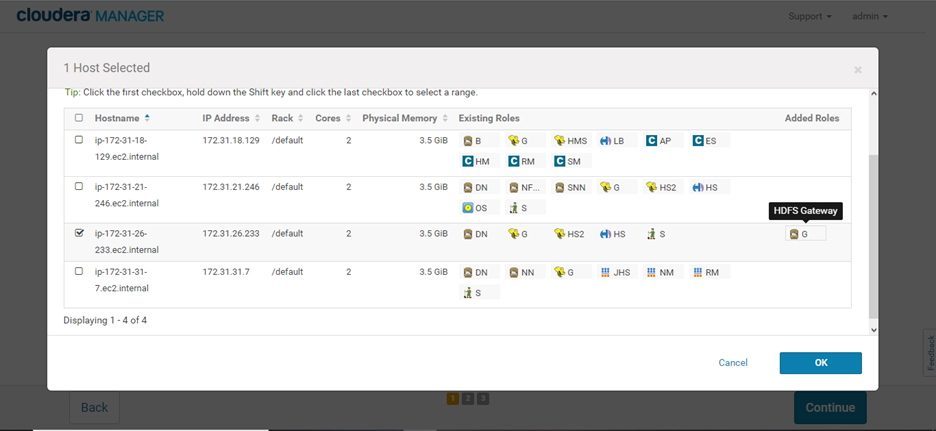
****

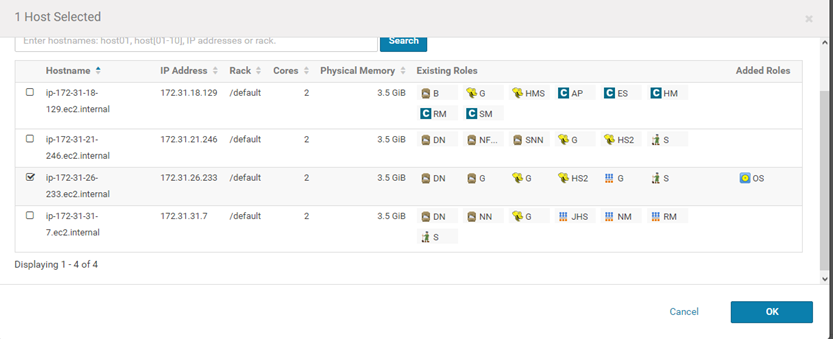
* **Cluster Details With Screenshots**

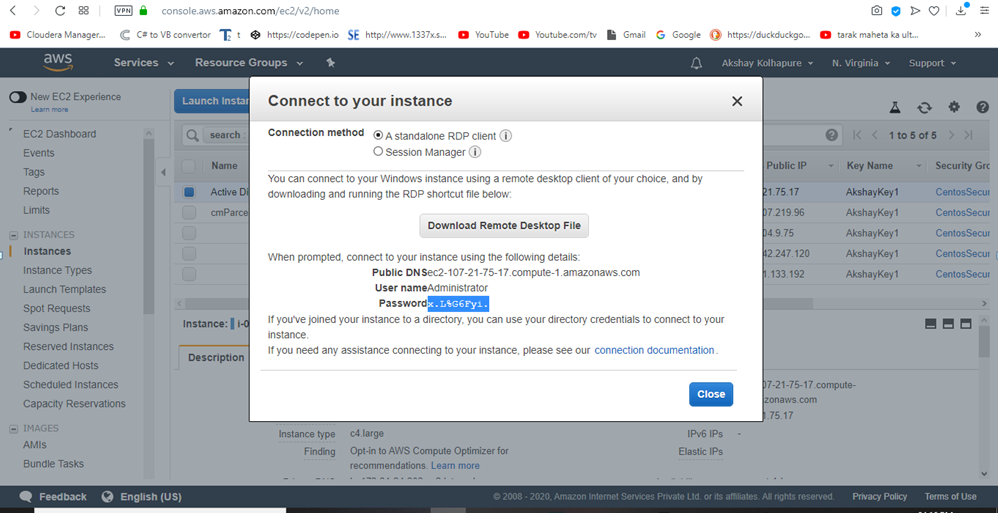
Here , We used Amazon Web Services (AWS) to built the cluster . We installed Cloudera Manager (CM) one node / instance ,installed all core services on nodes with the help of **‘Deploy Like A Boss’** manner. Following are some screenshots of use case implementation,

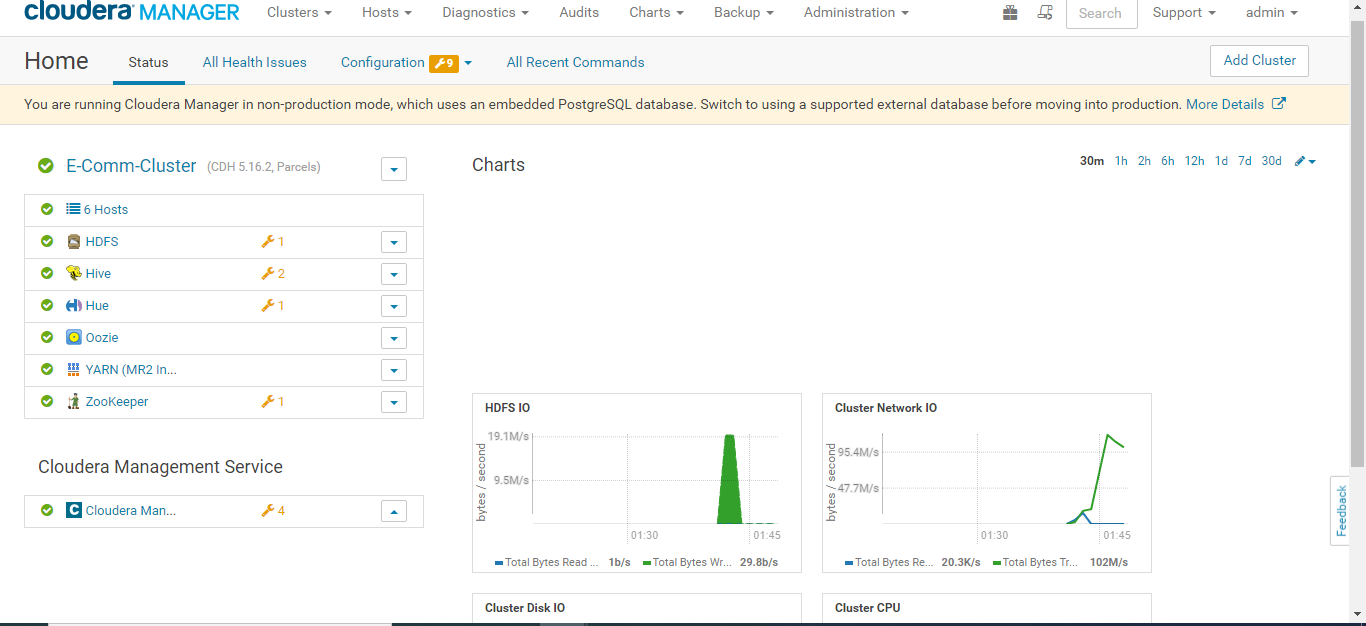
1. **Instances**

****

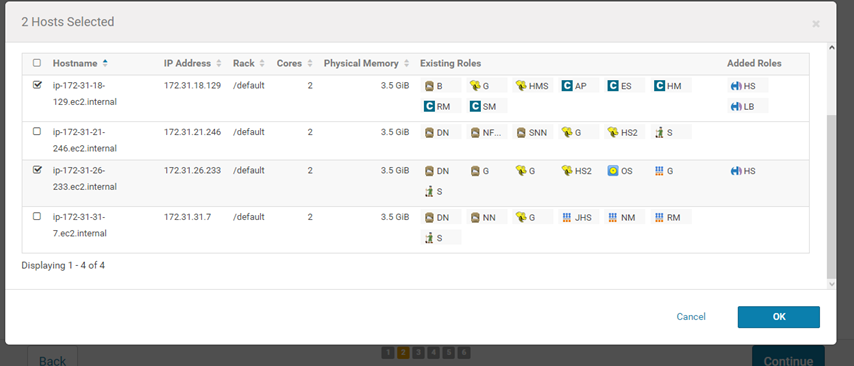
****

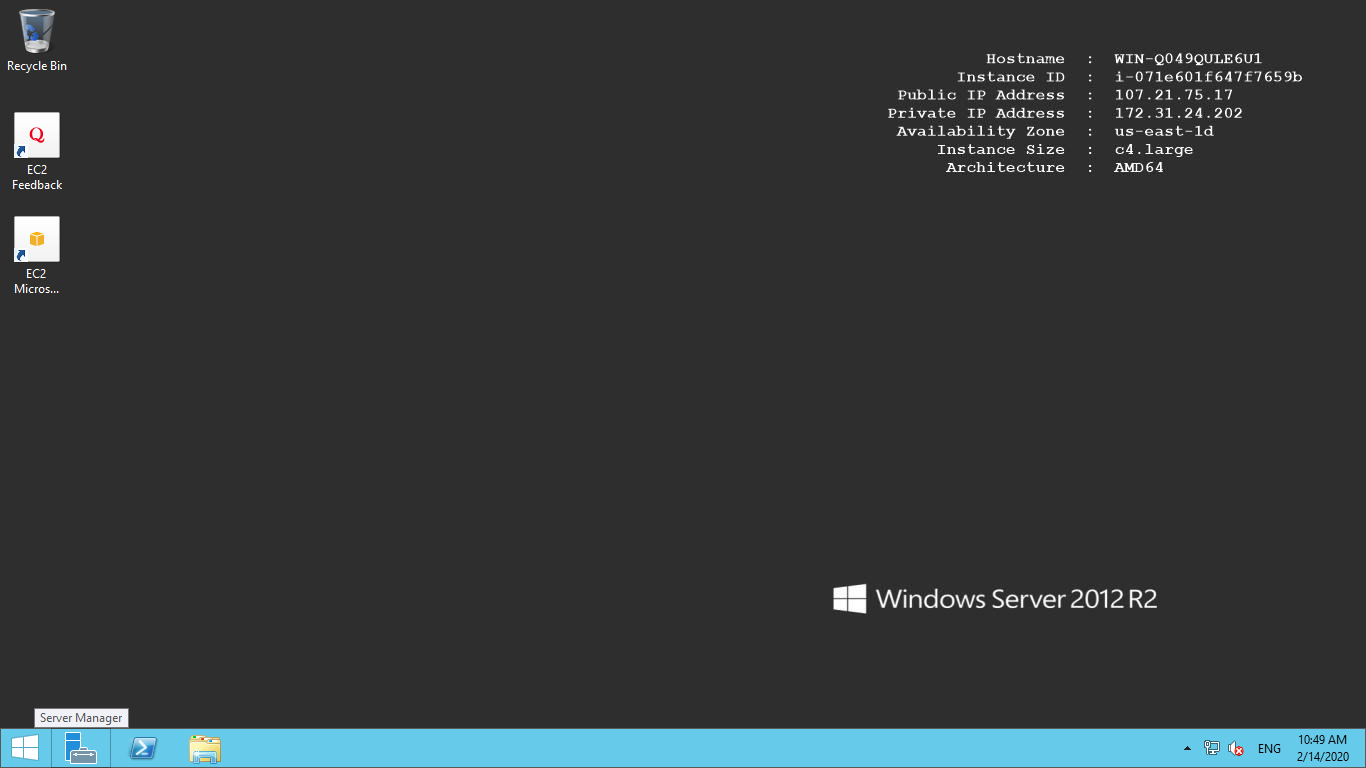
****

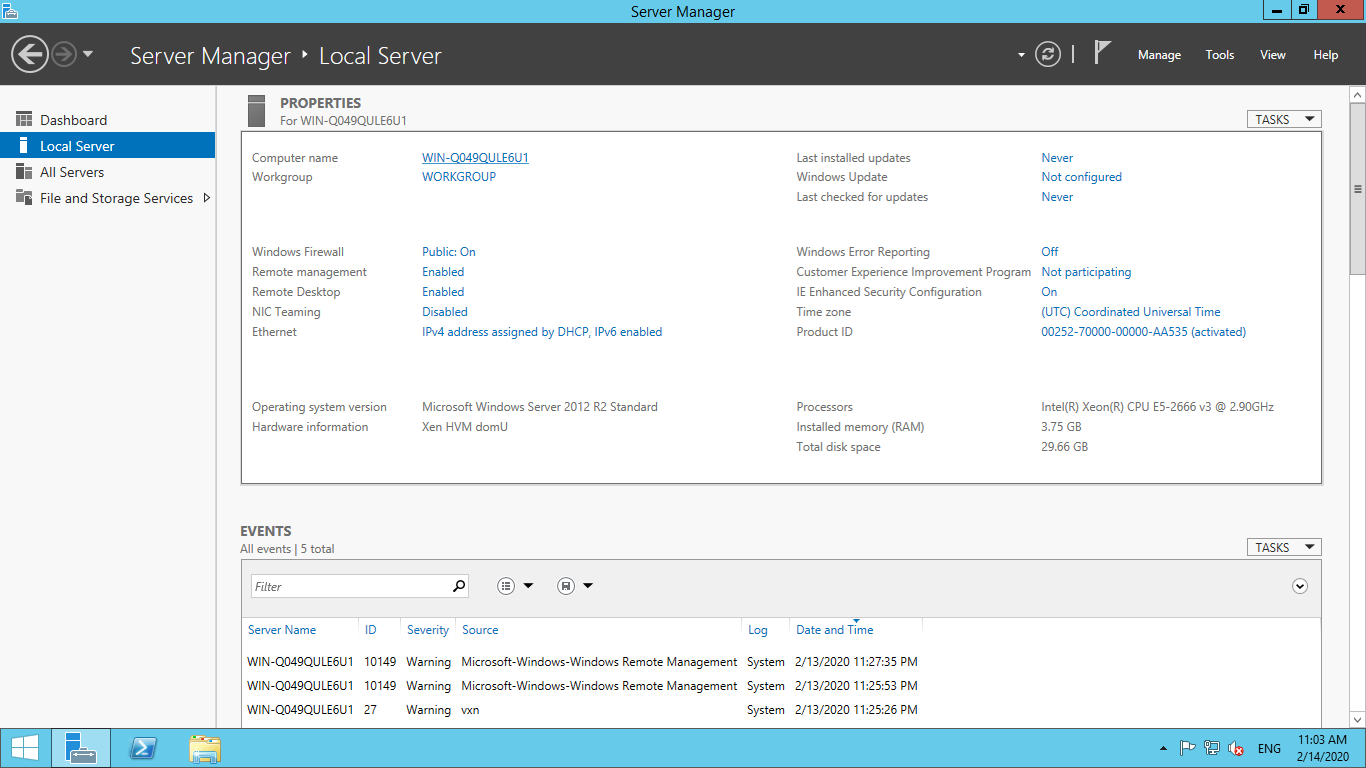
****

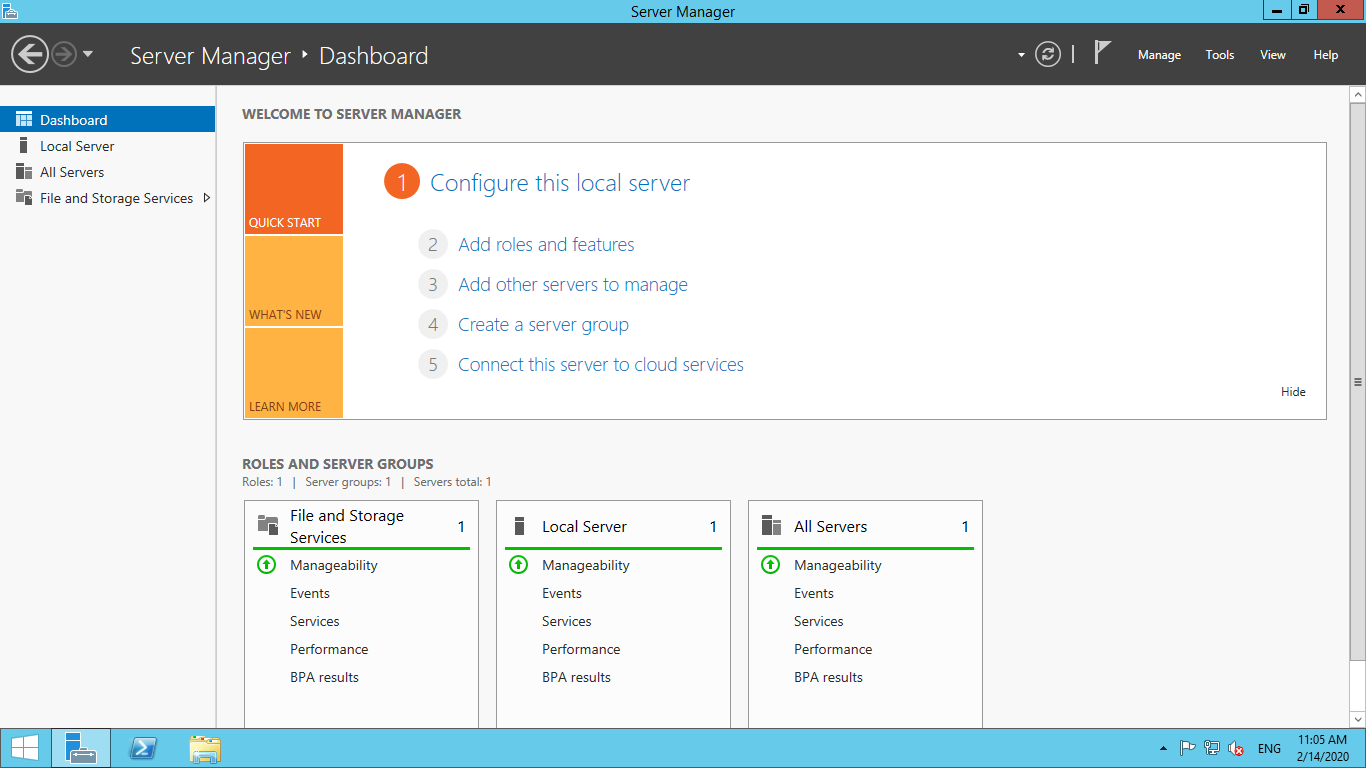
****

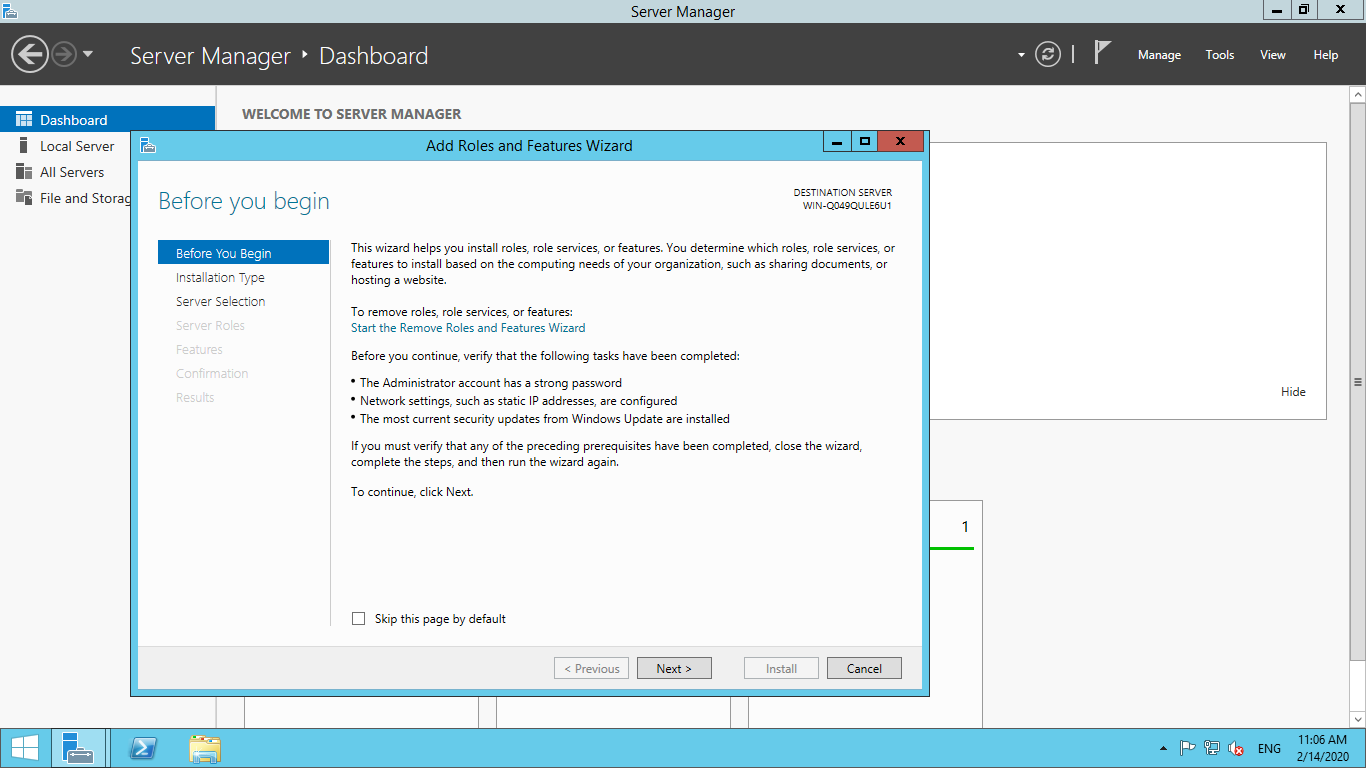
* **Installed HUE**

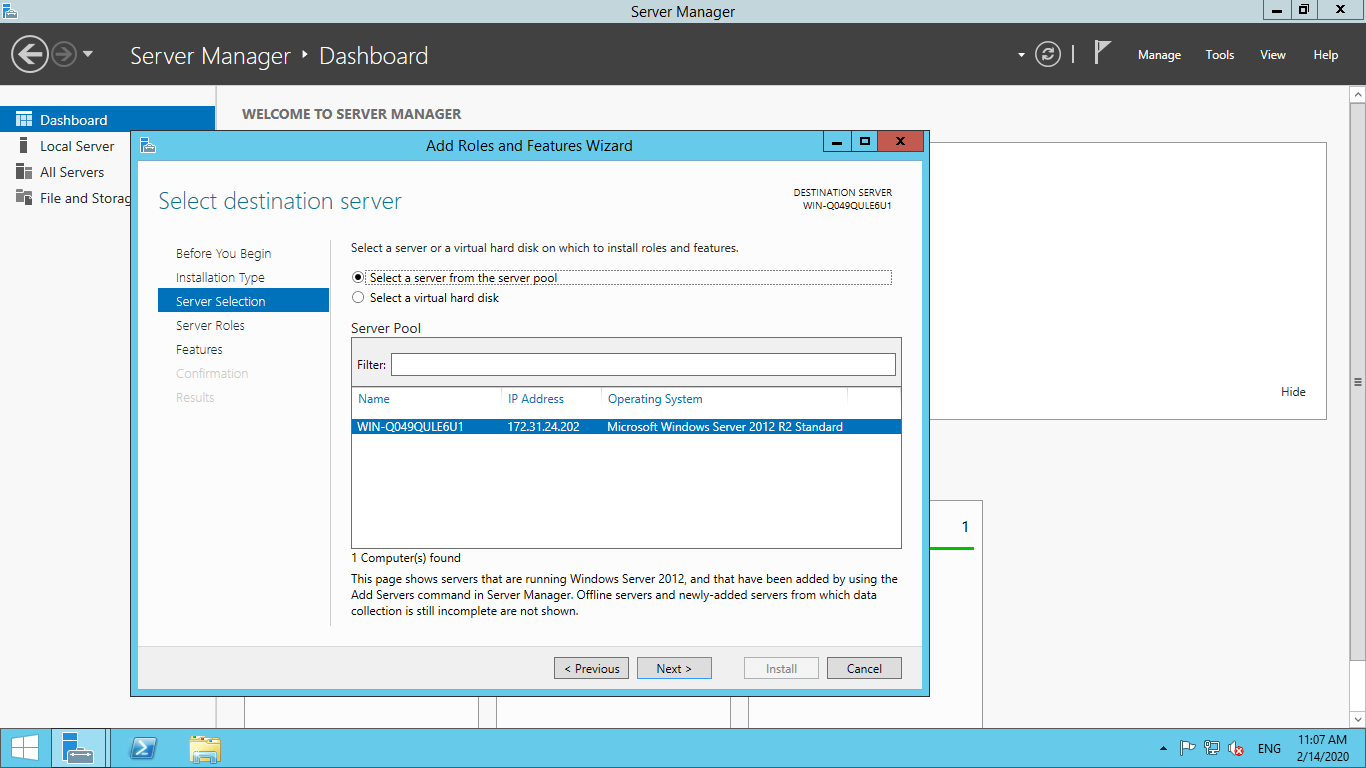


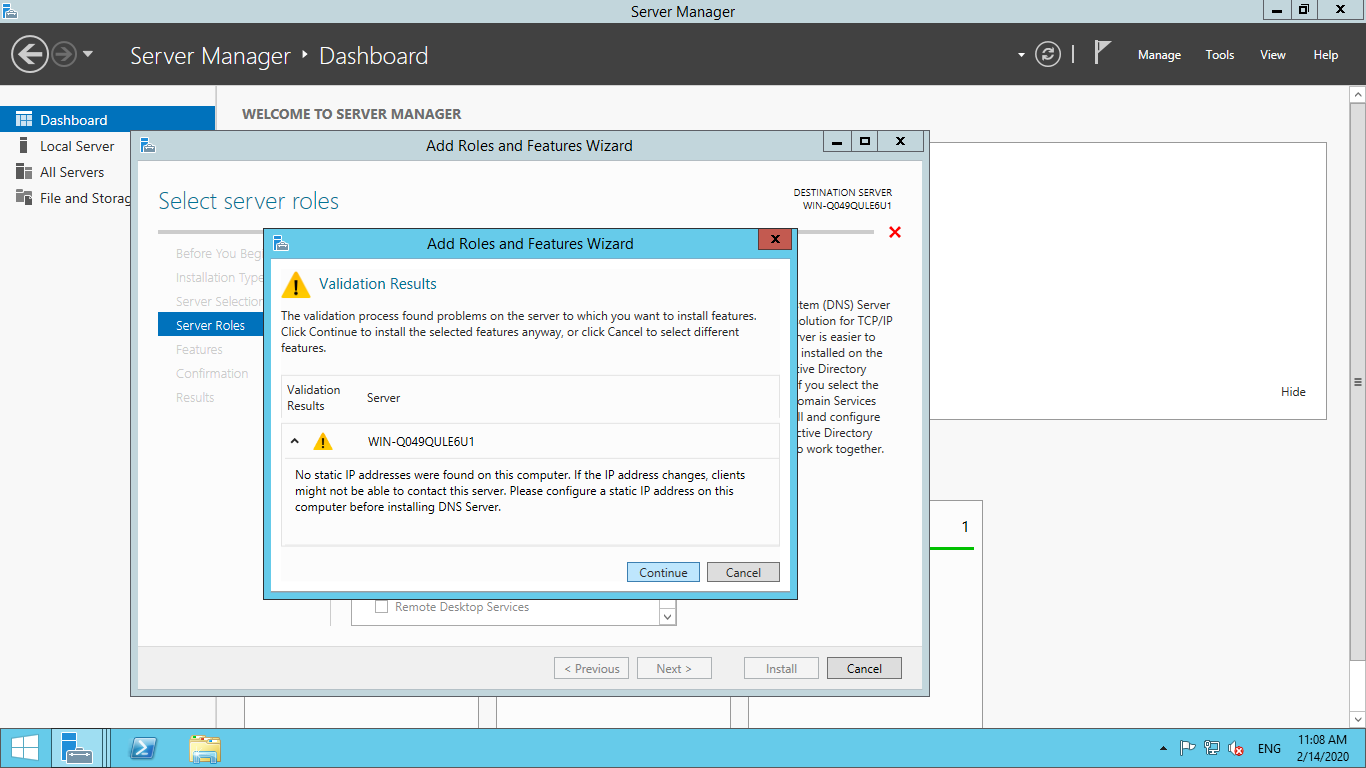




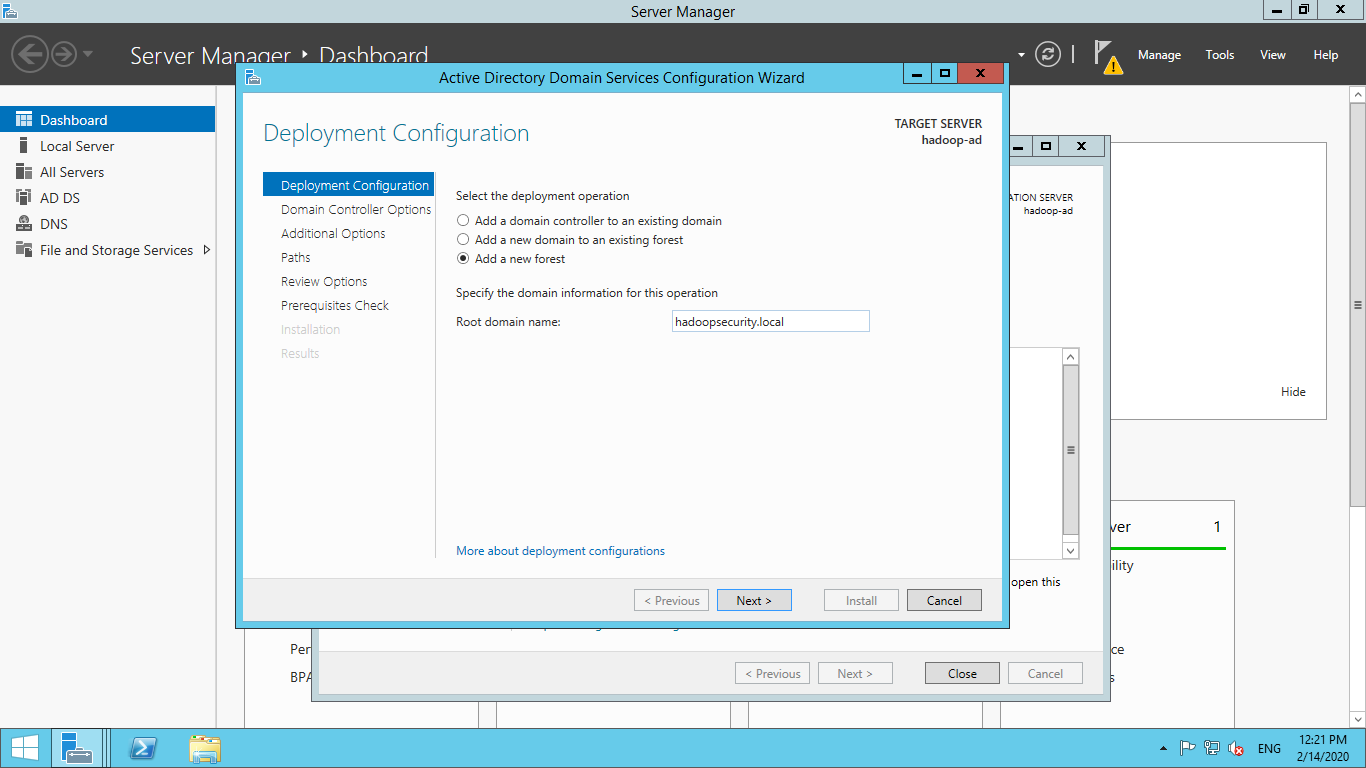


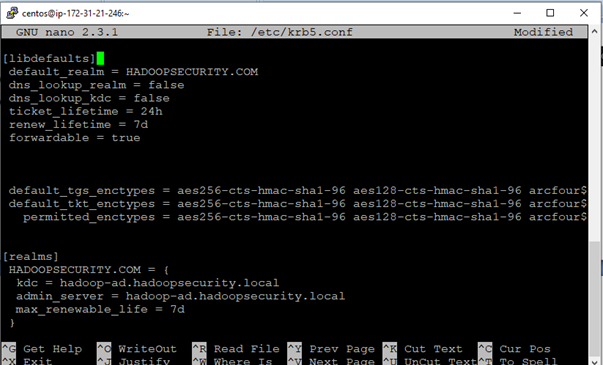


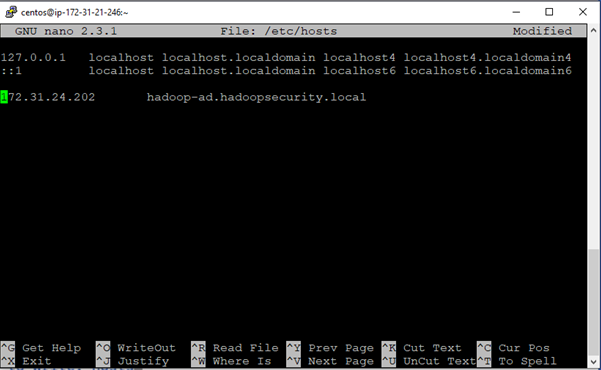


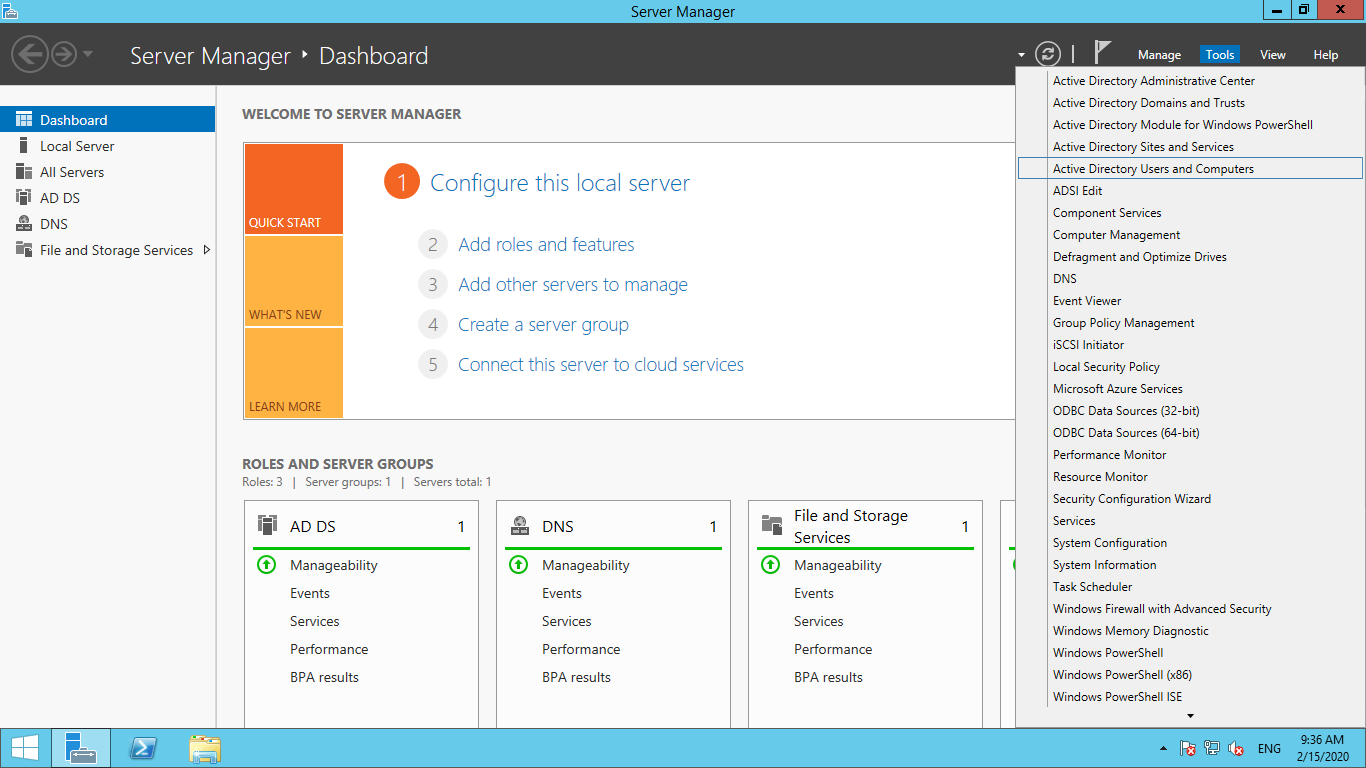


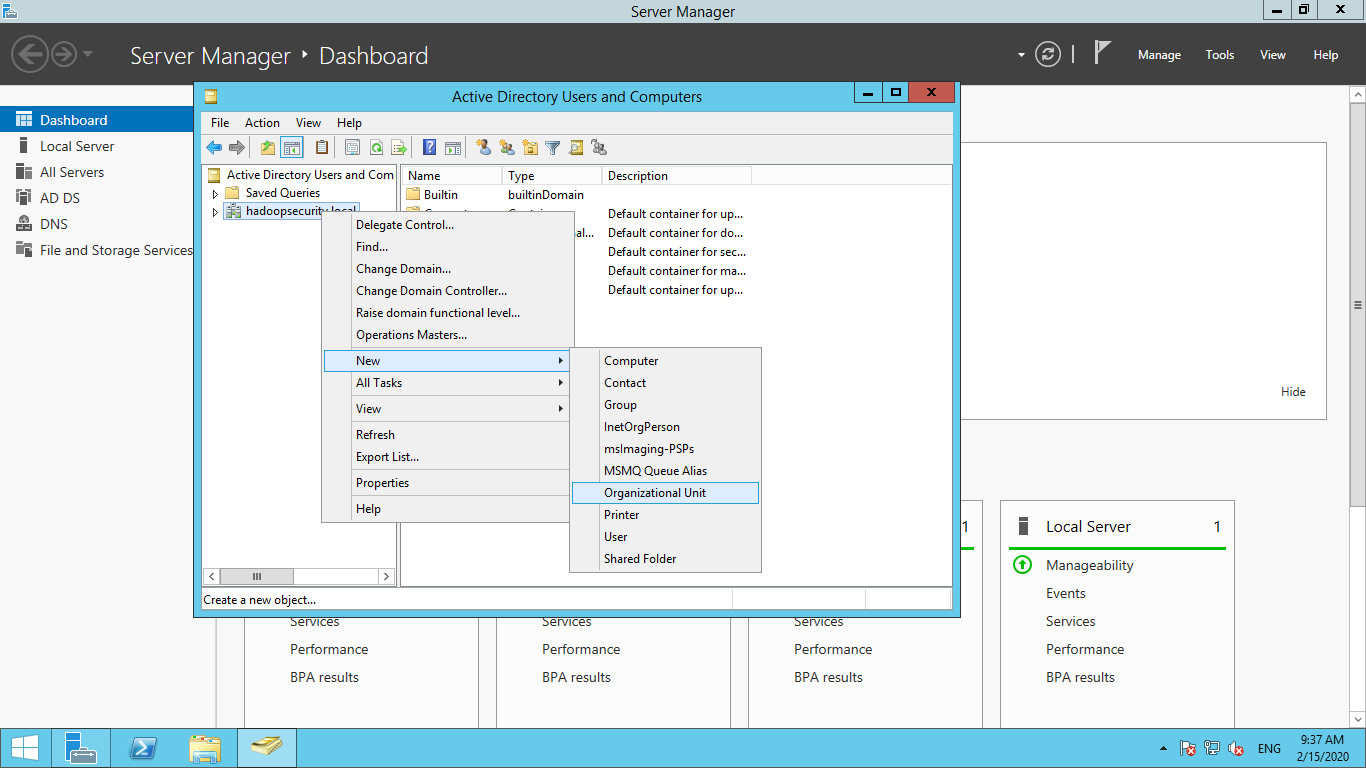


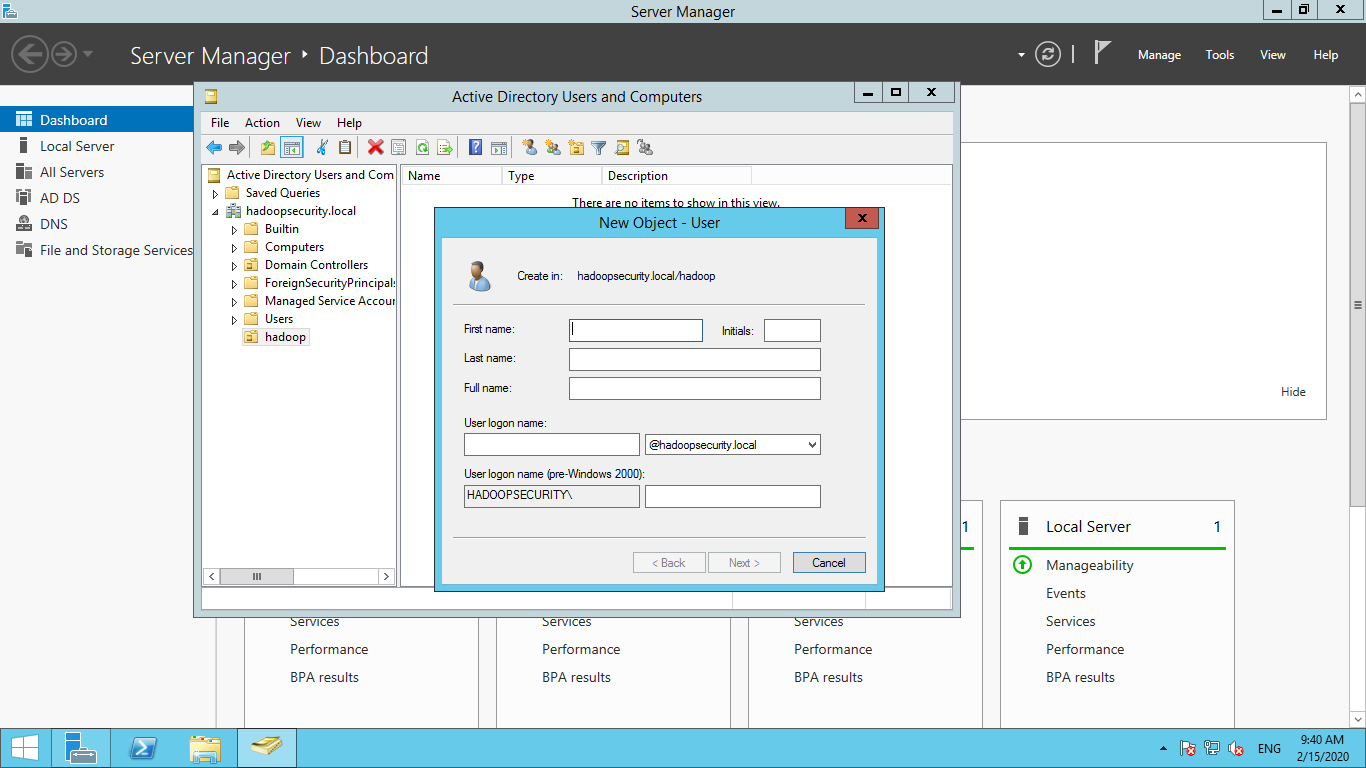


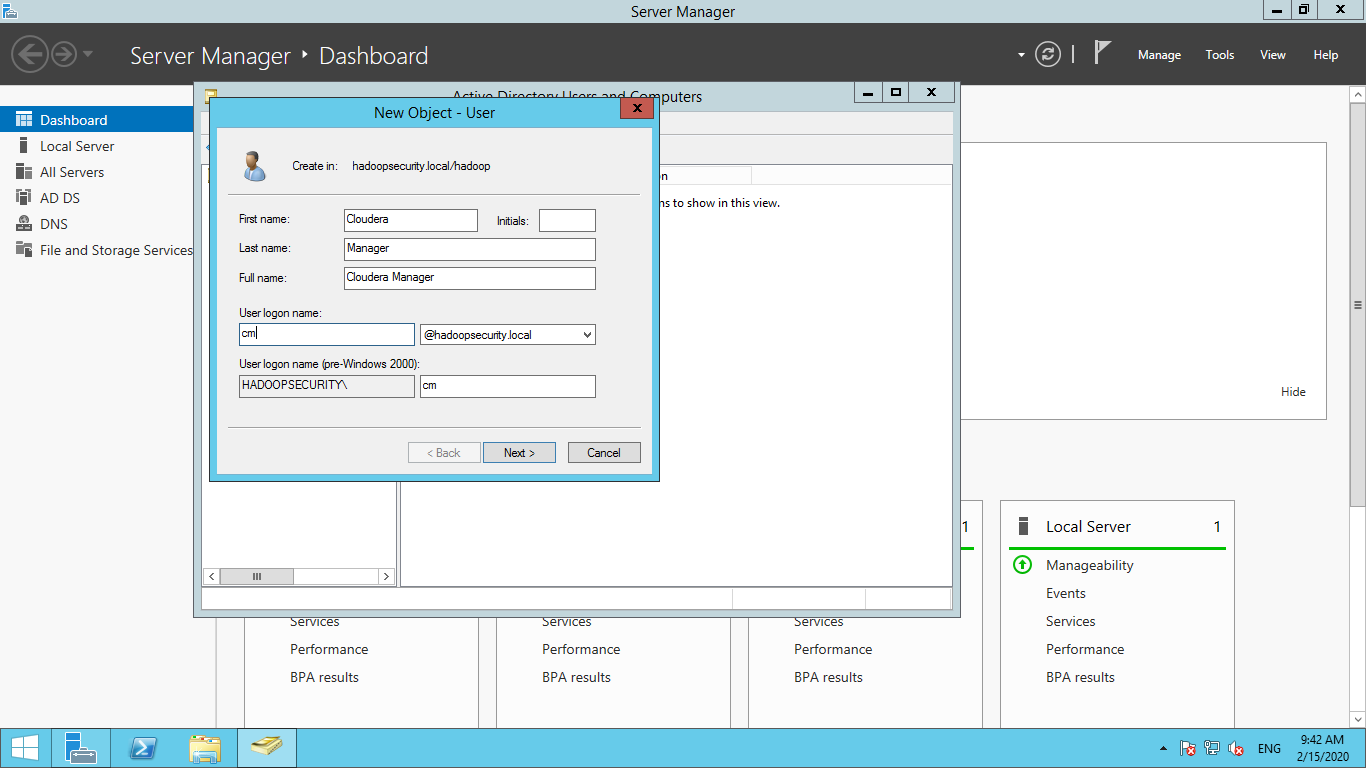


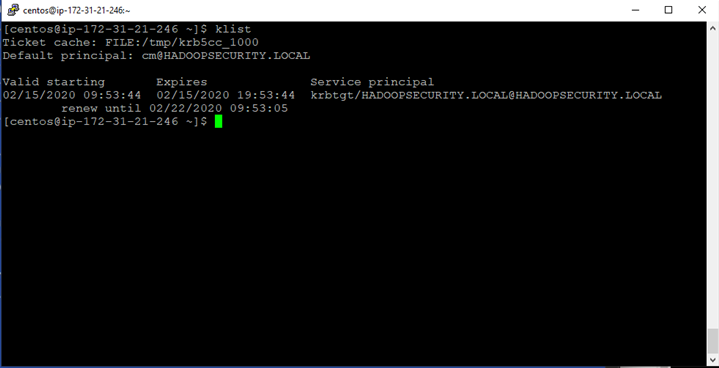


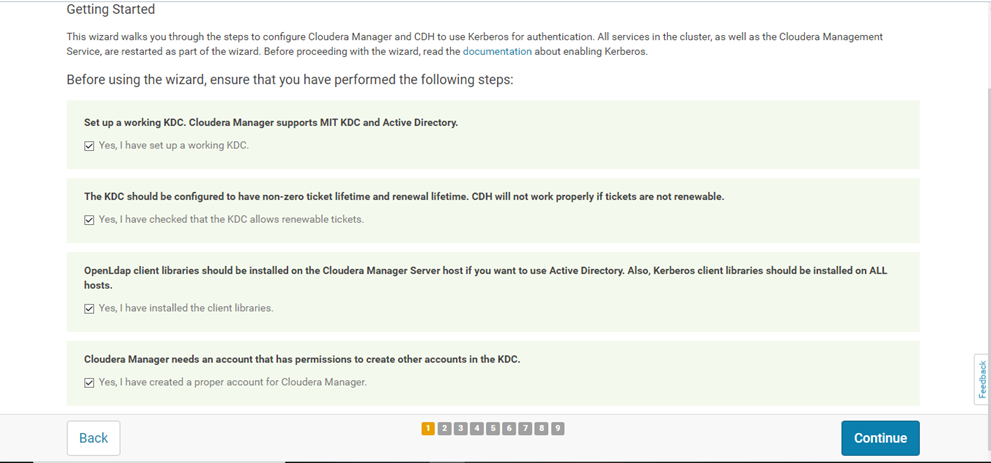


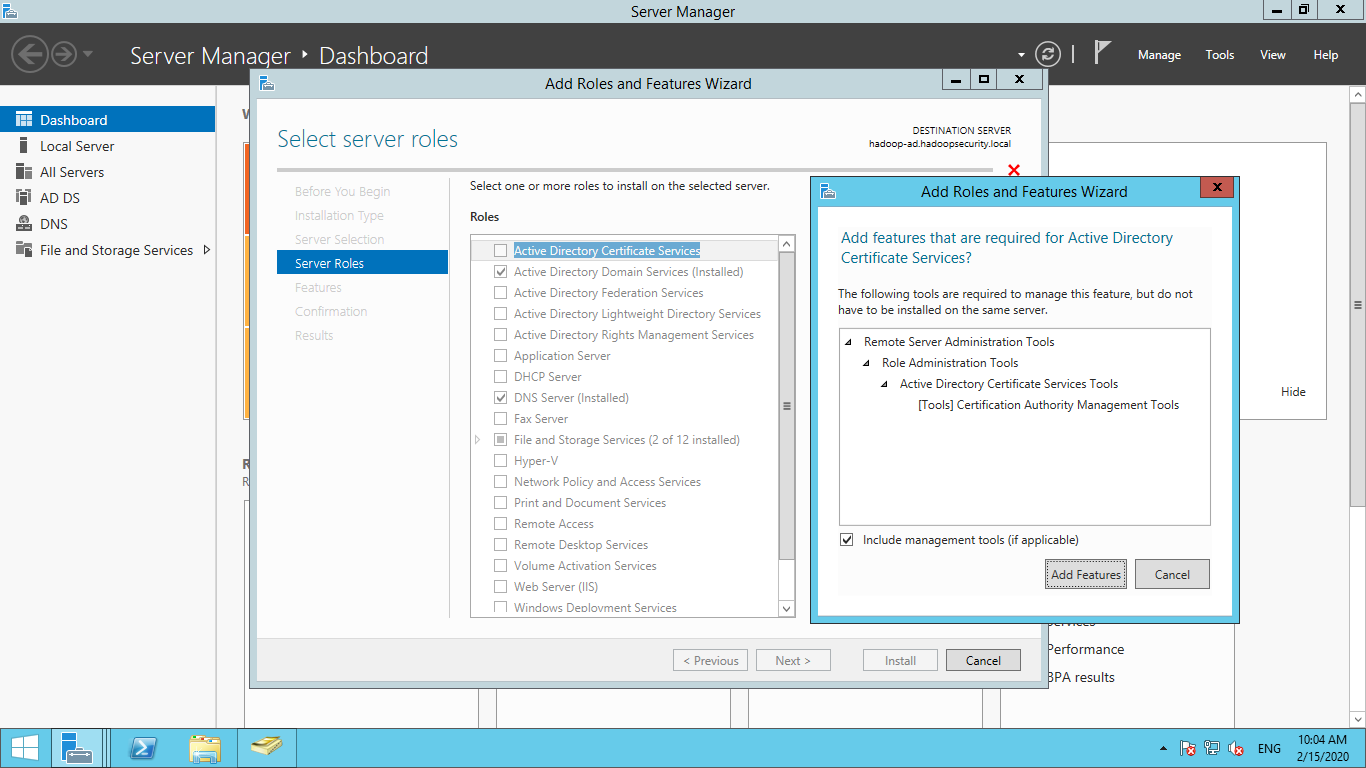


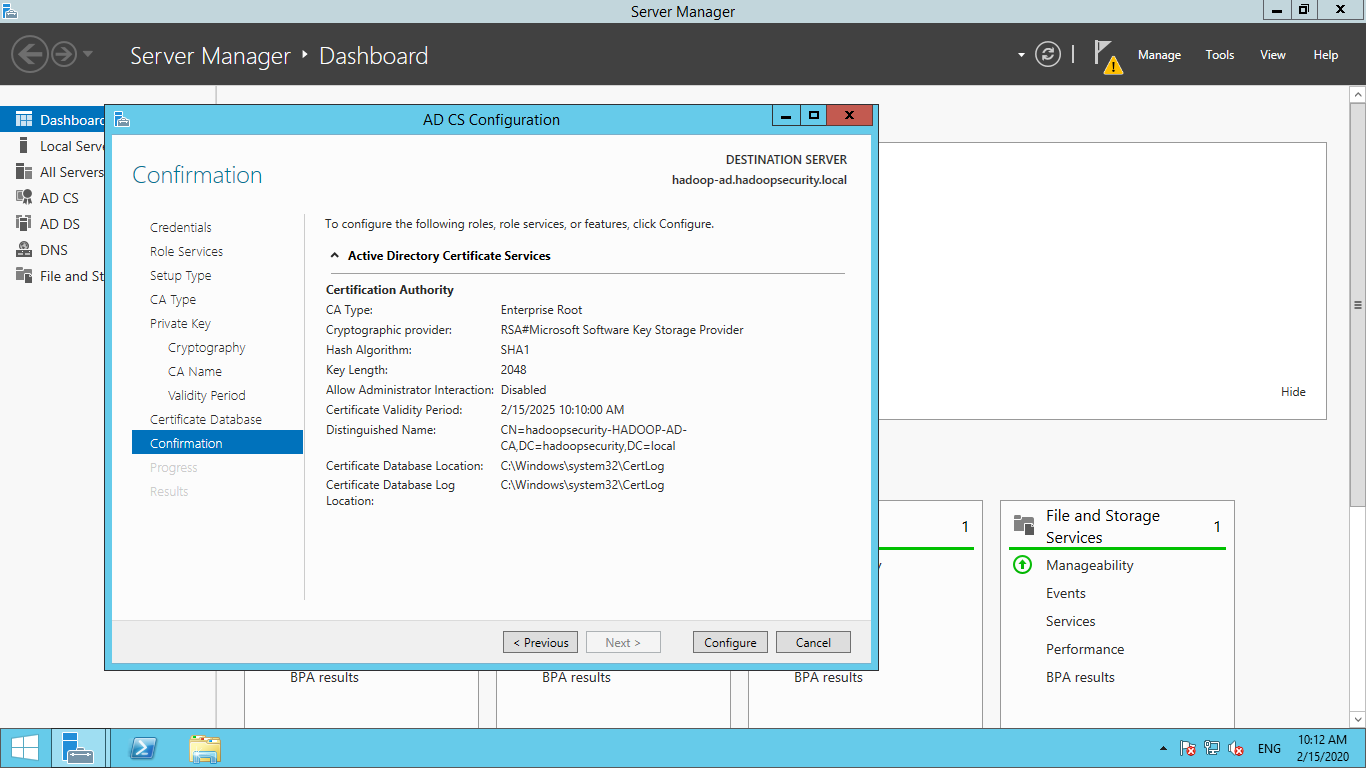


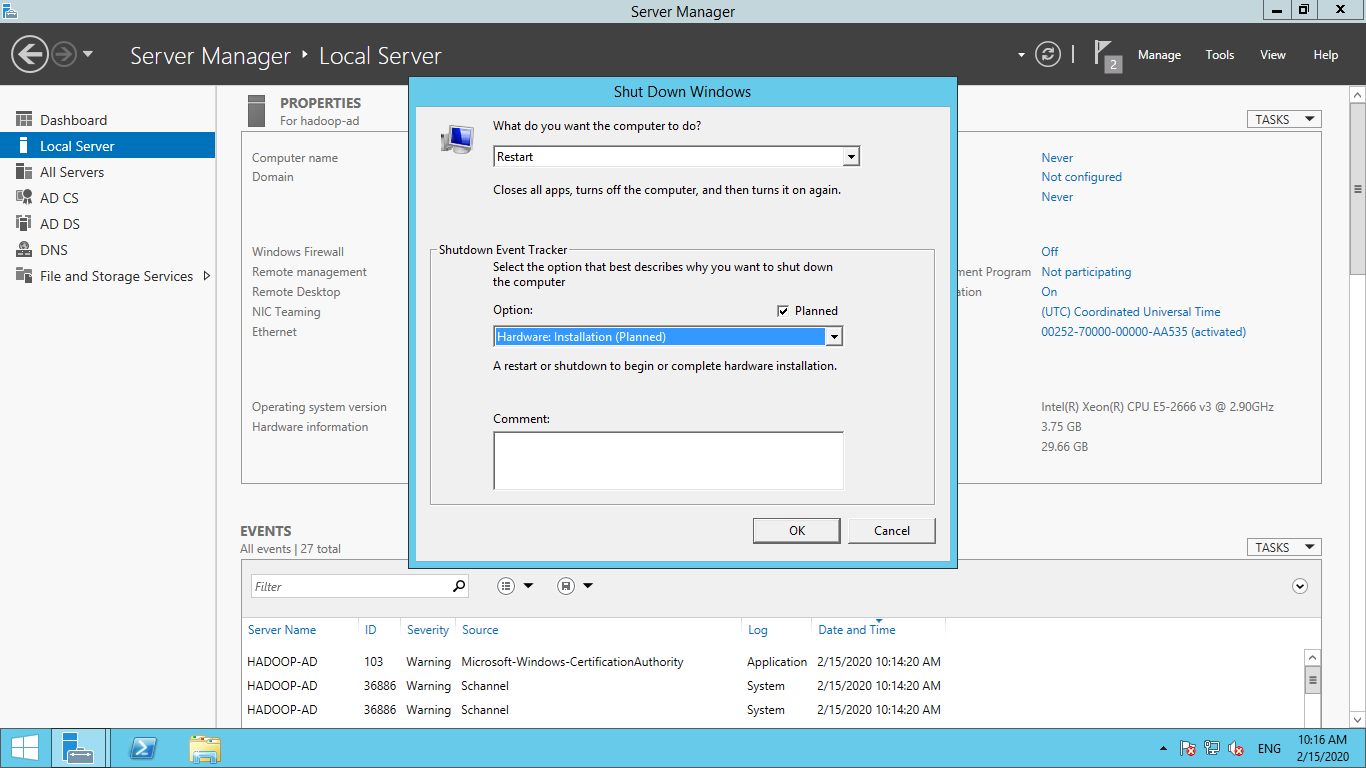


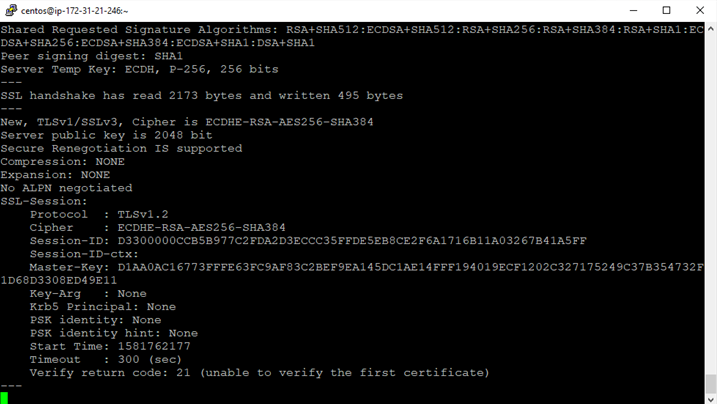


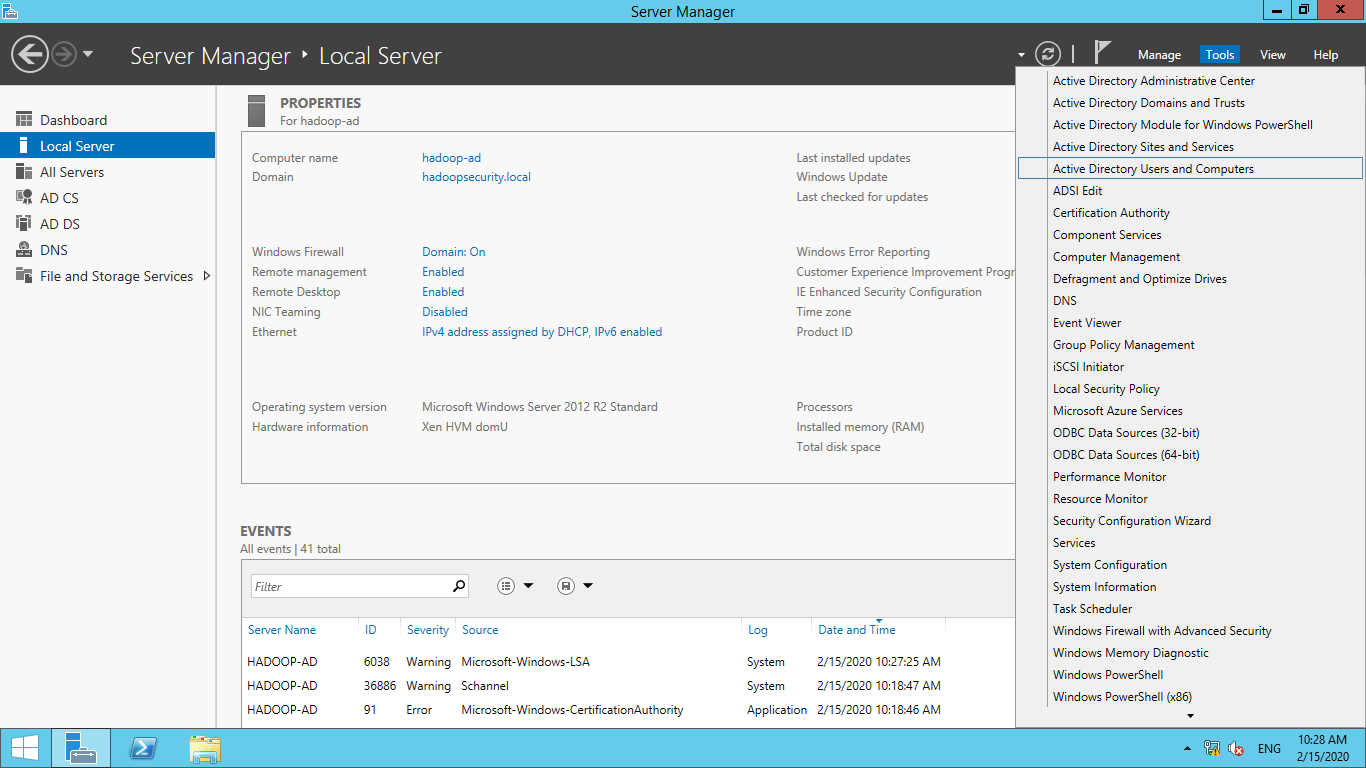


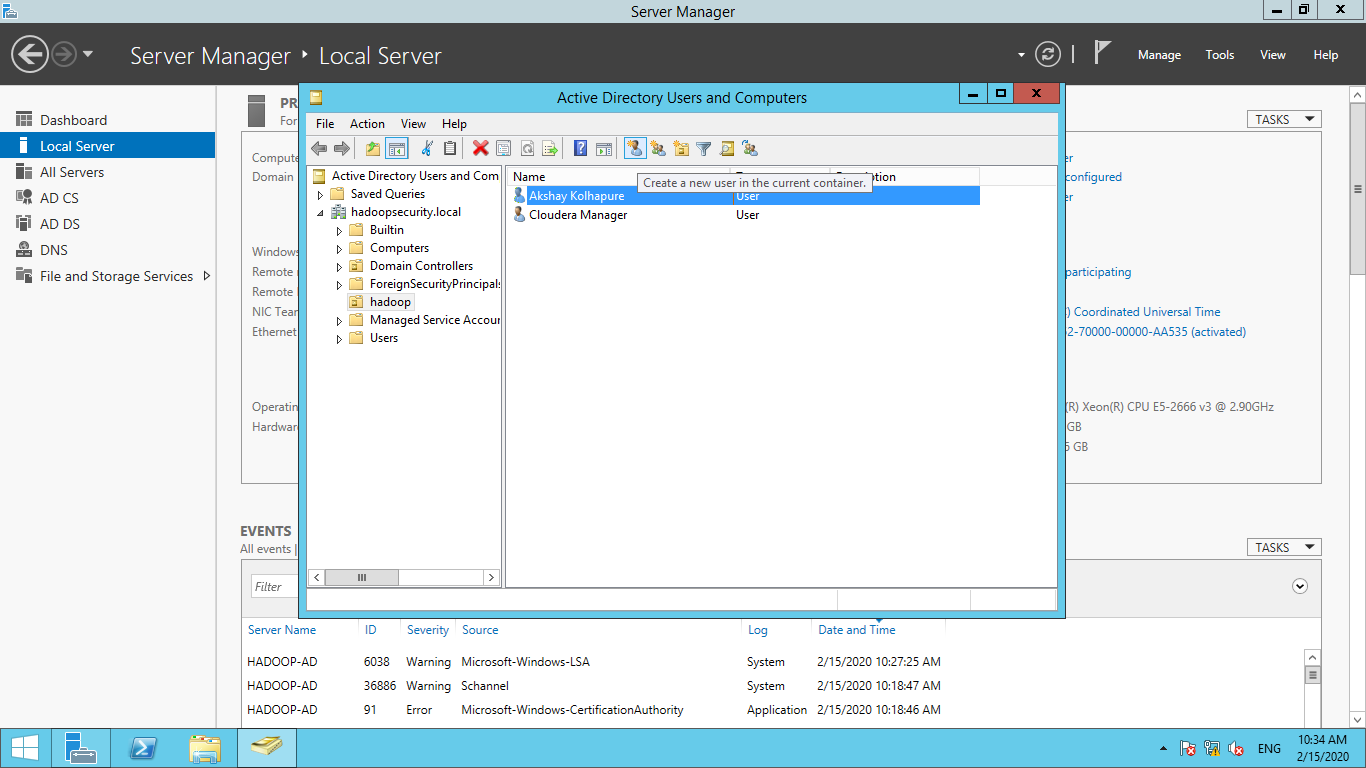


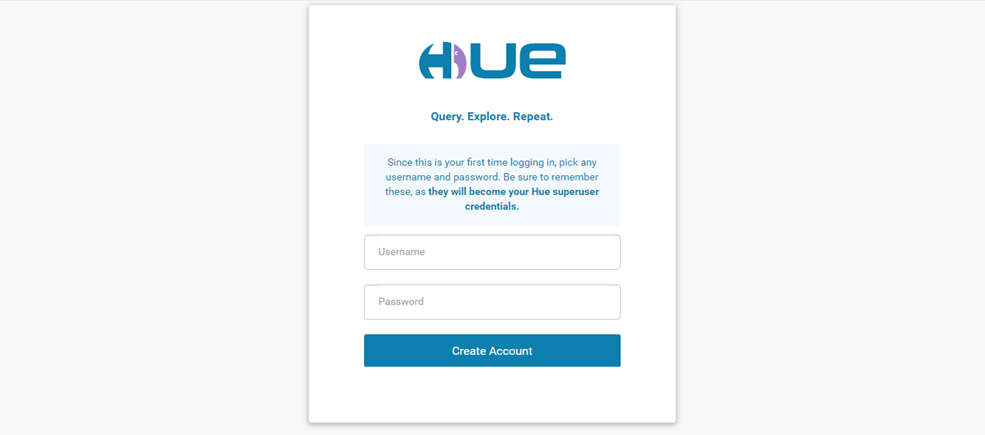


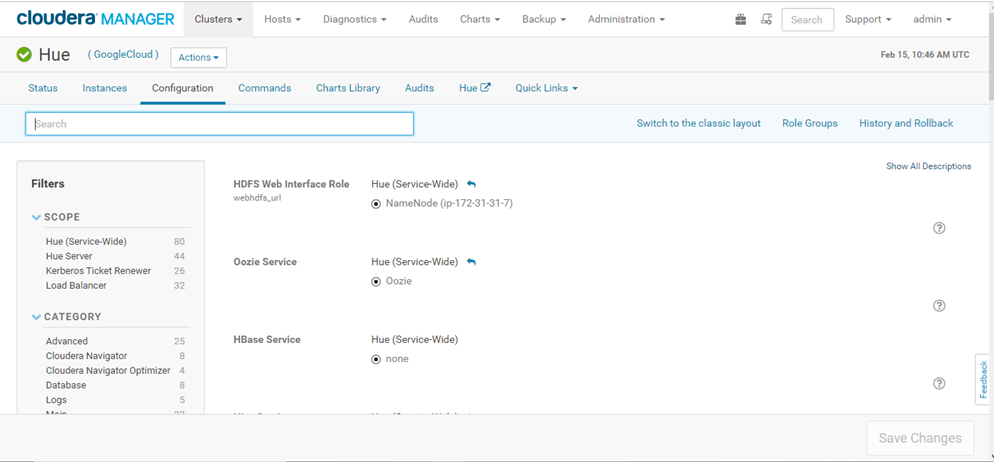


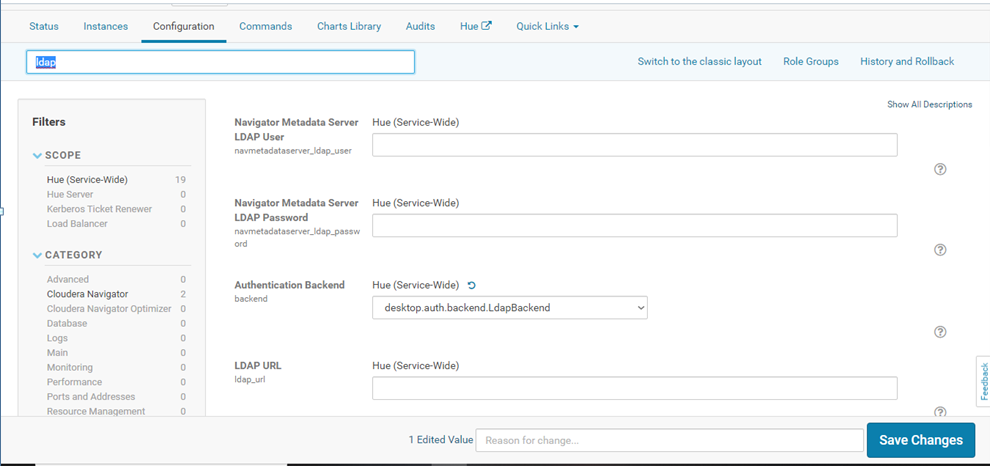


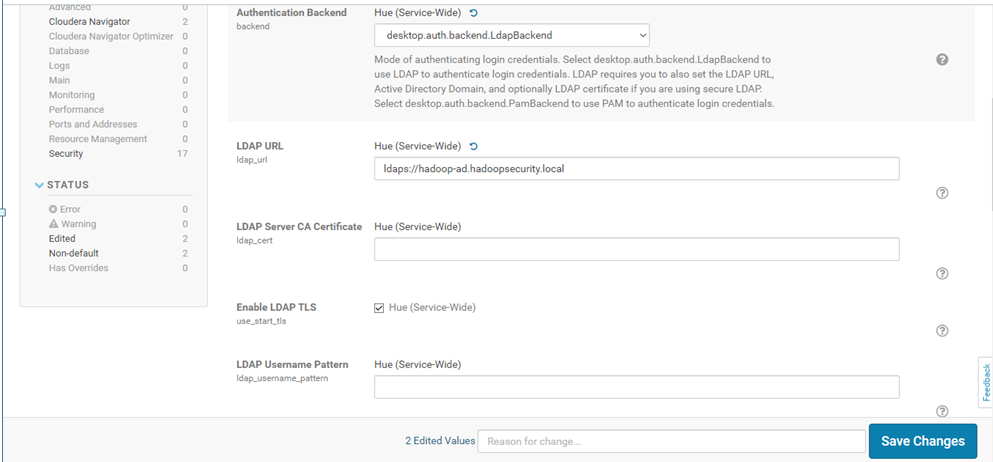


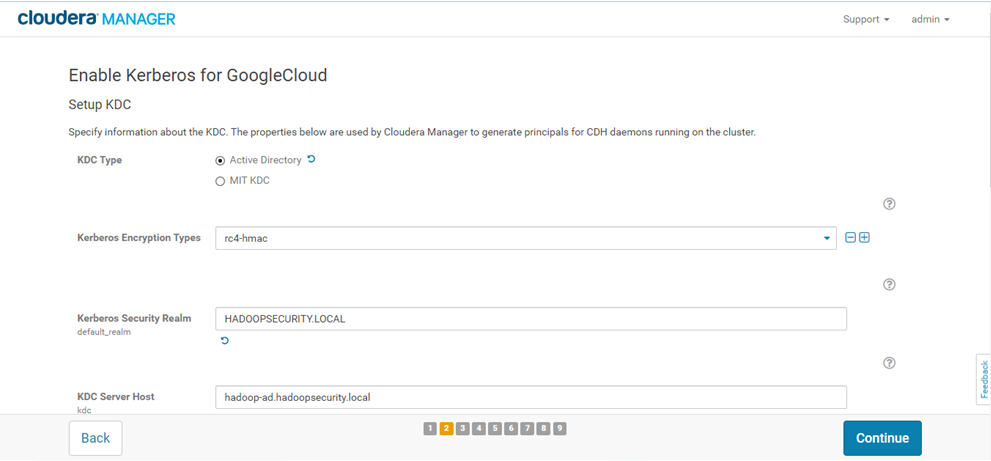


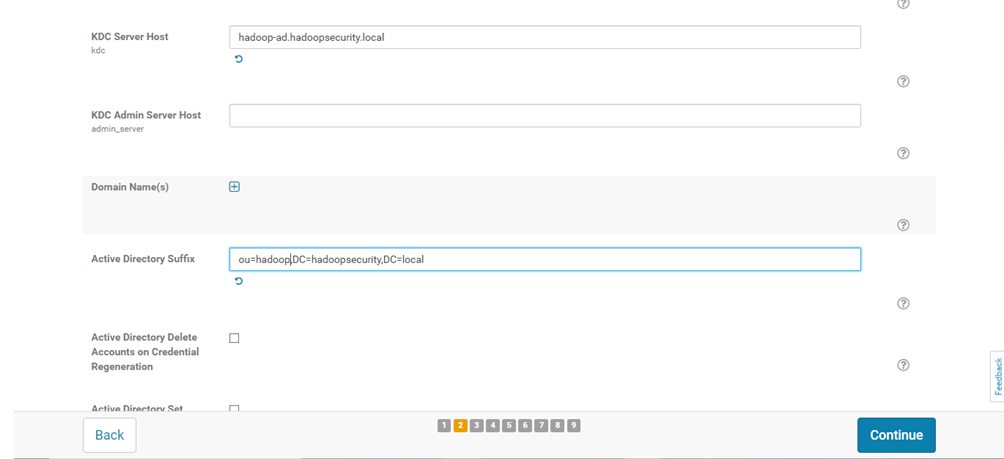


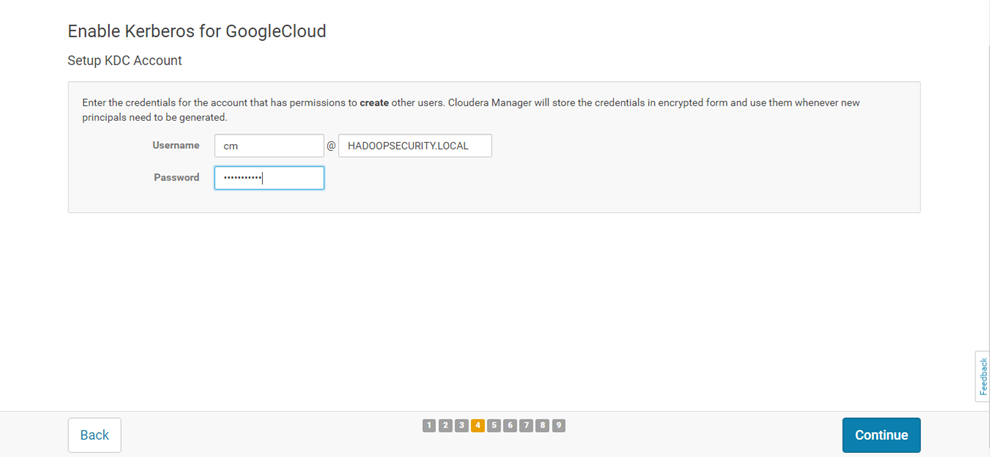


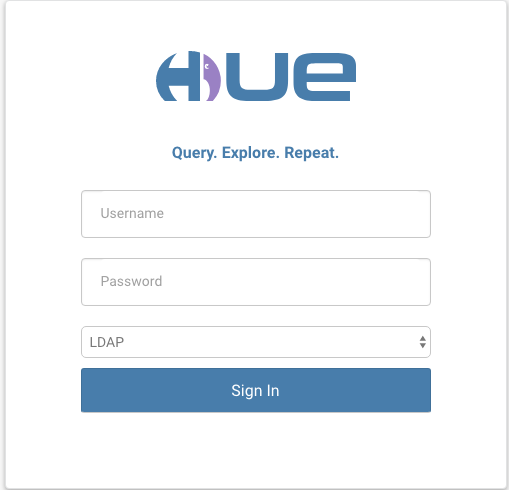




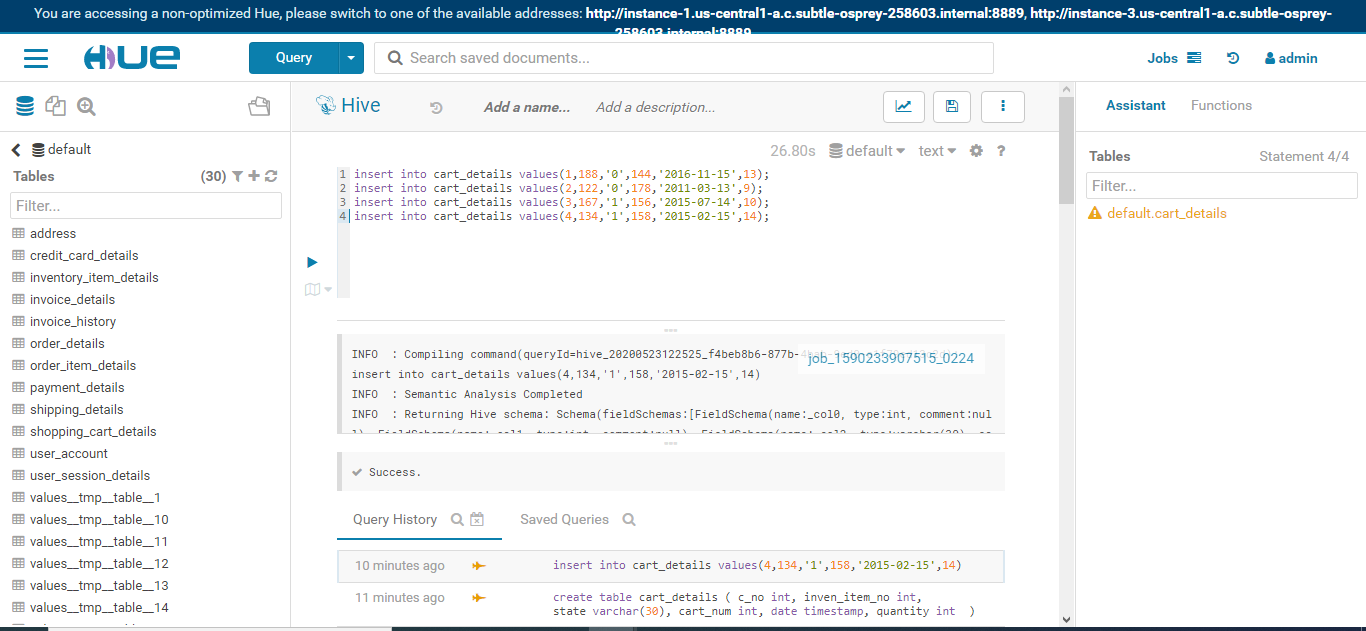


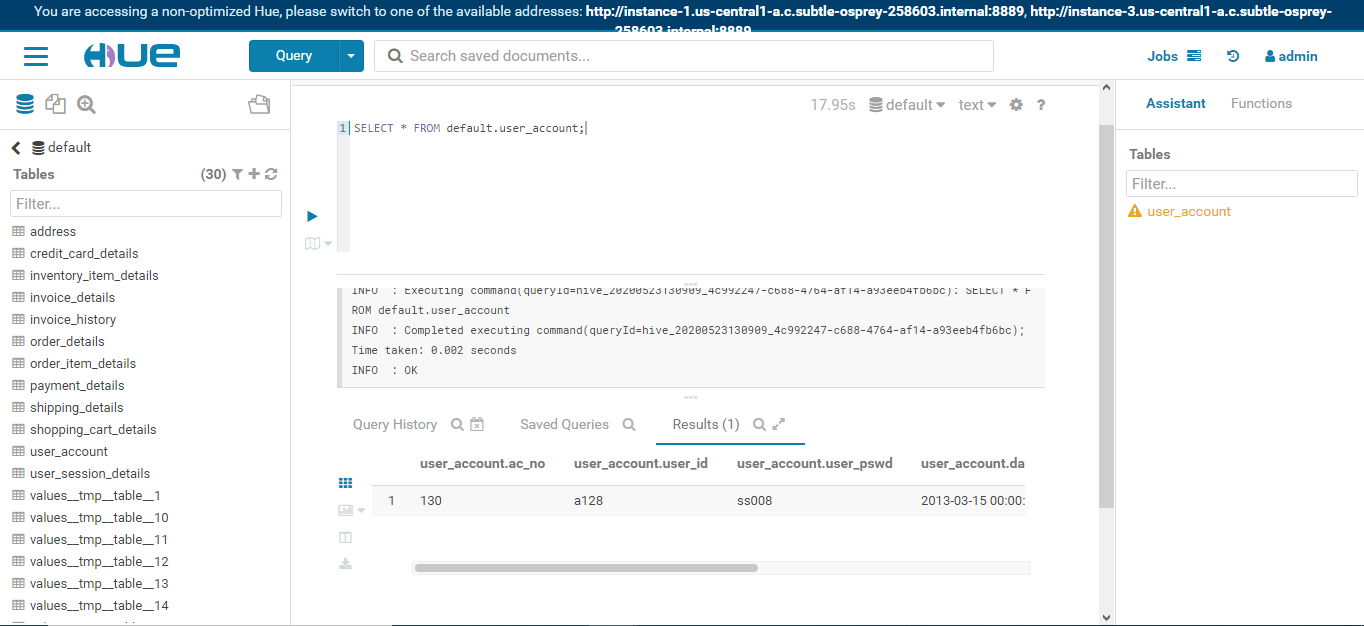


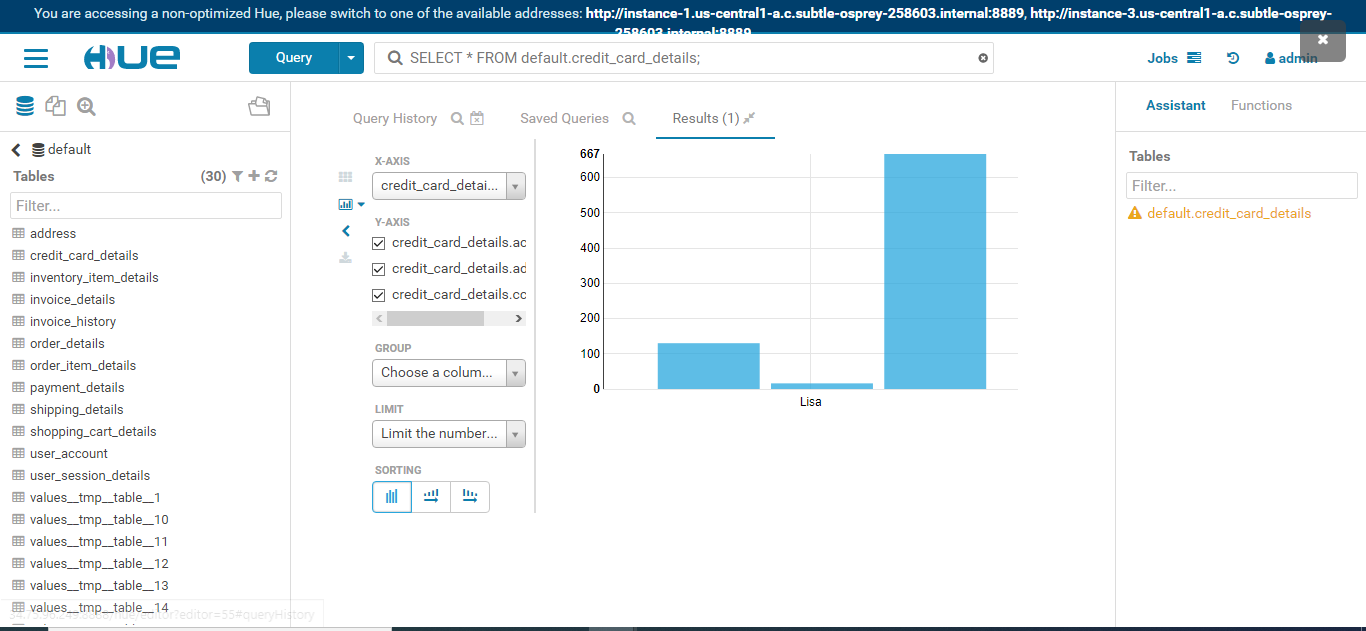


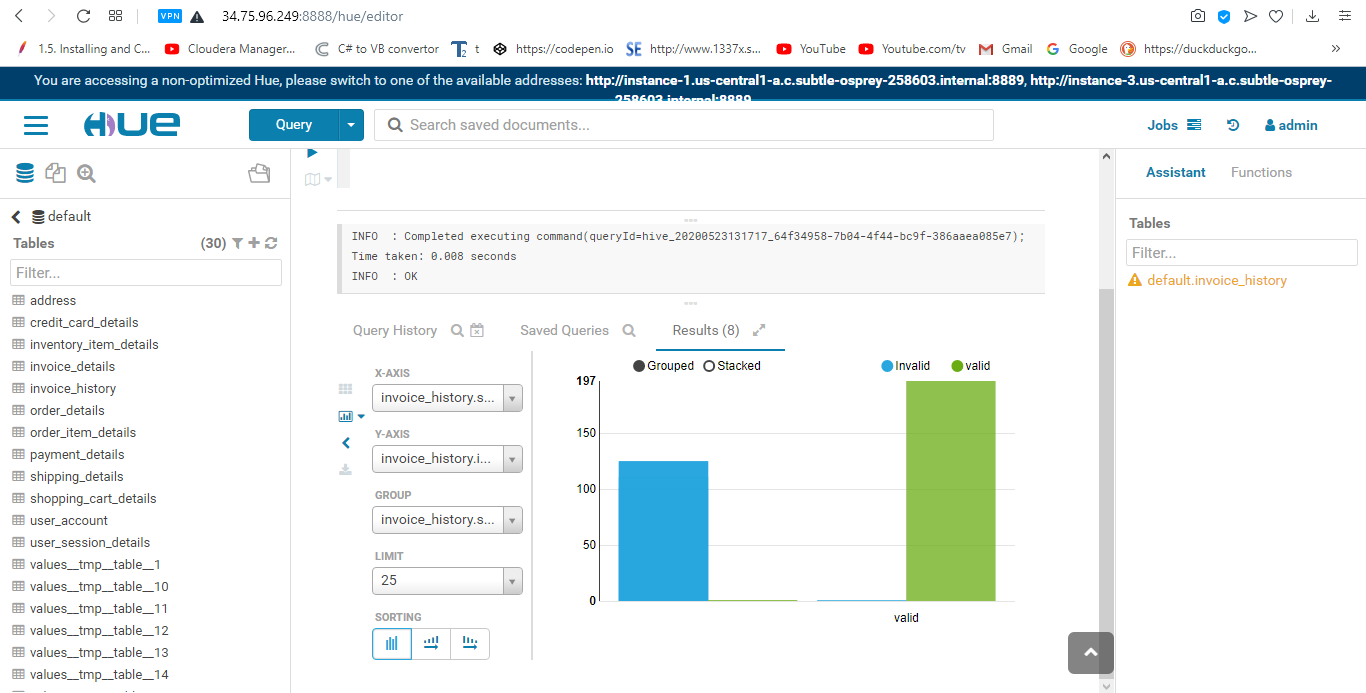


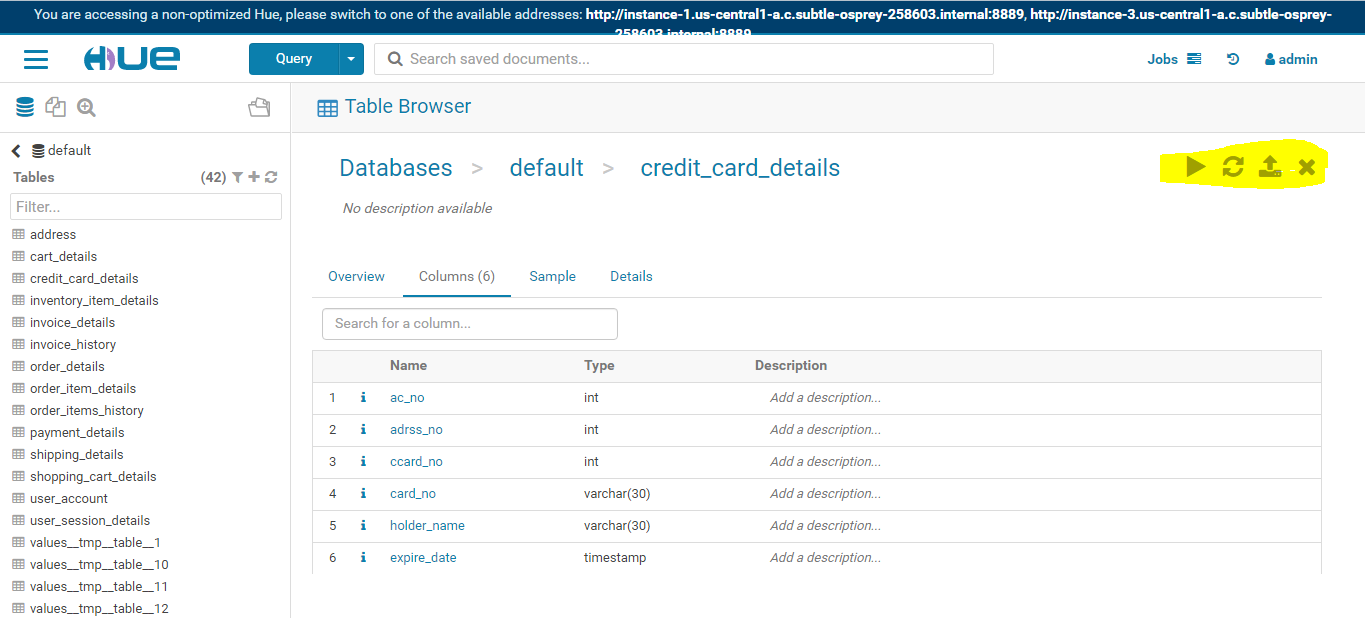
* **Hue Browser for HIVE user**

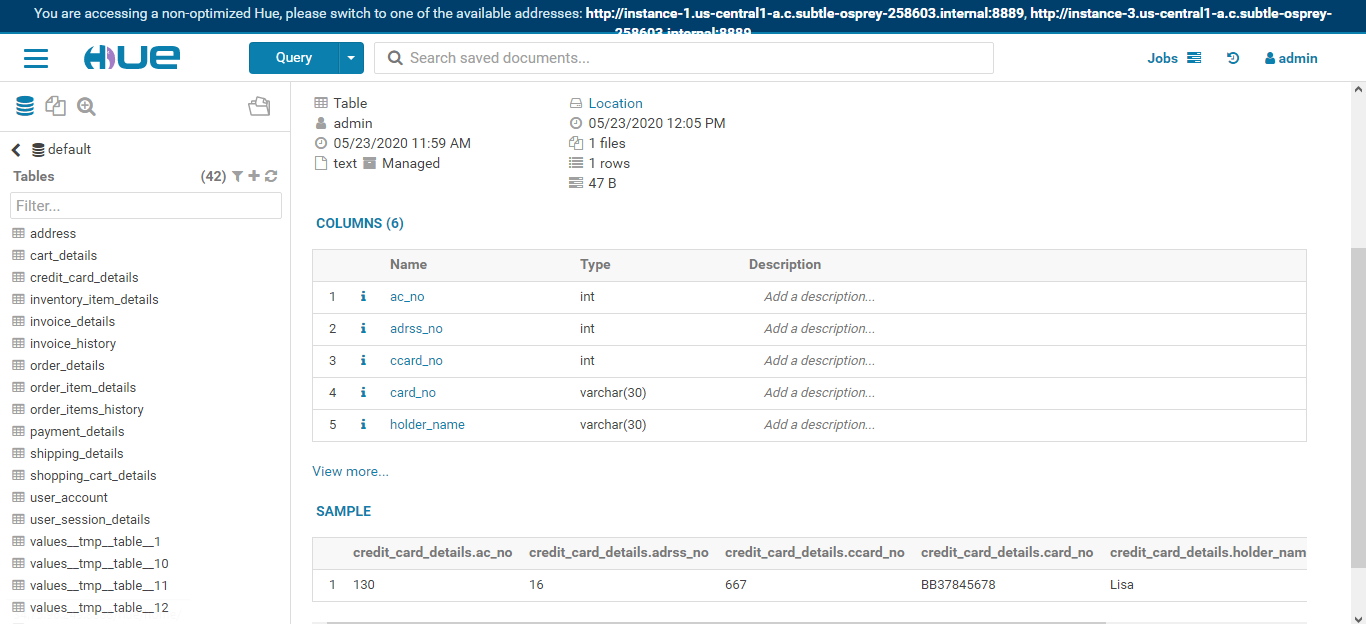


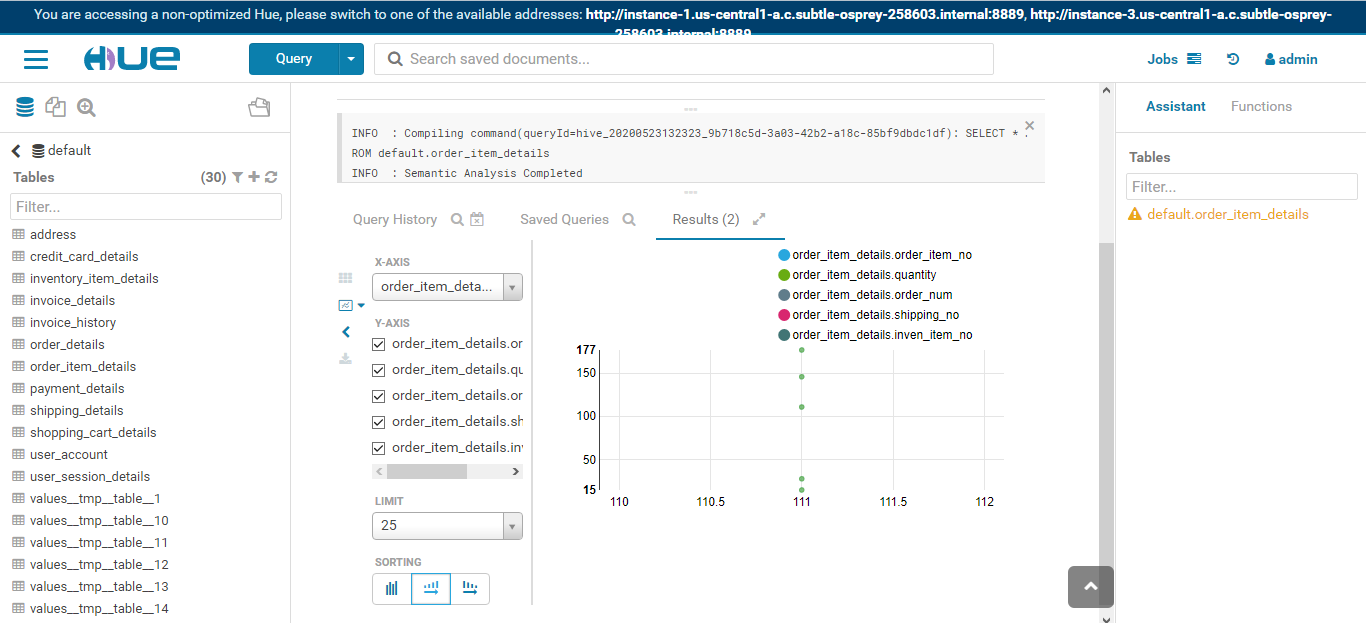
****

****

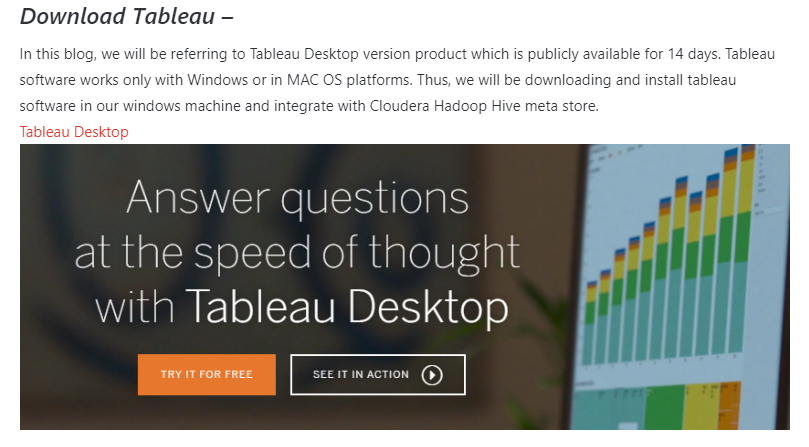
****

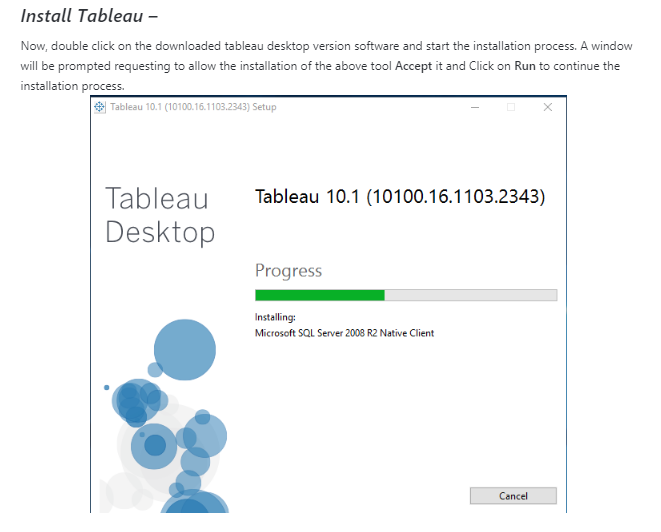
****

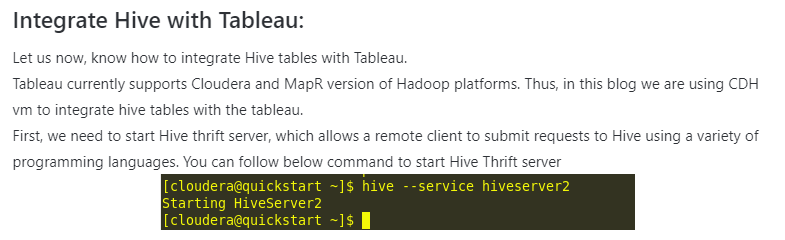
****

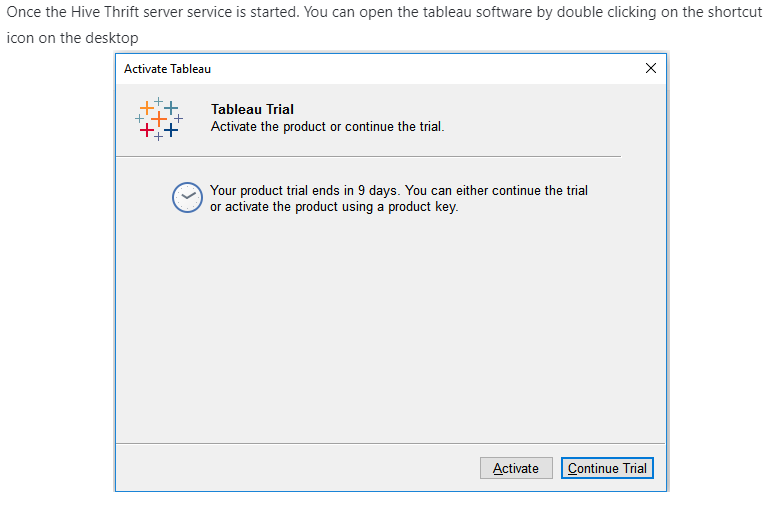
****

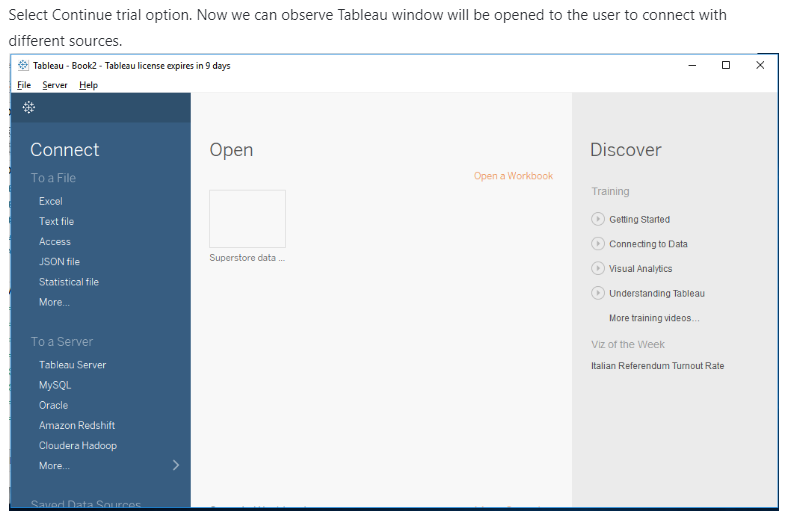
* **Tableau Connection**

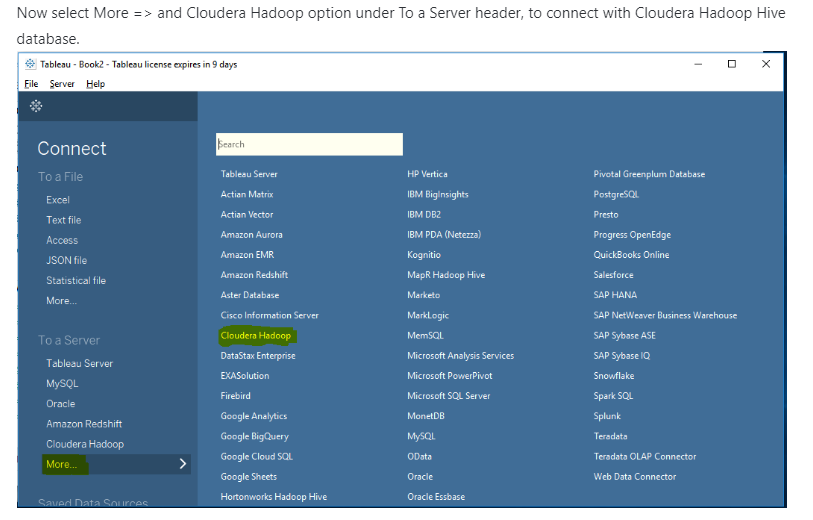
****

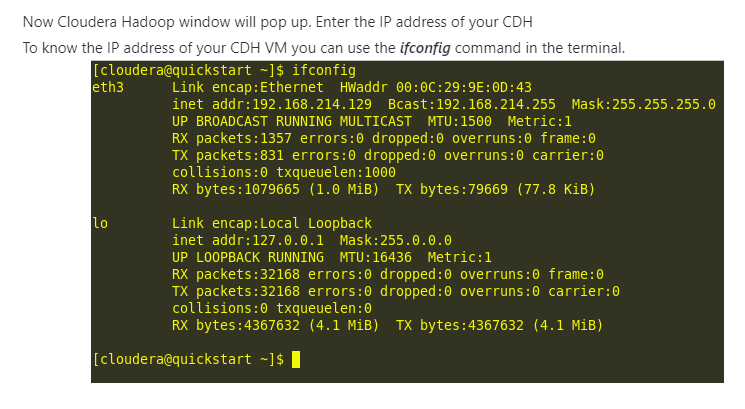
****

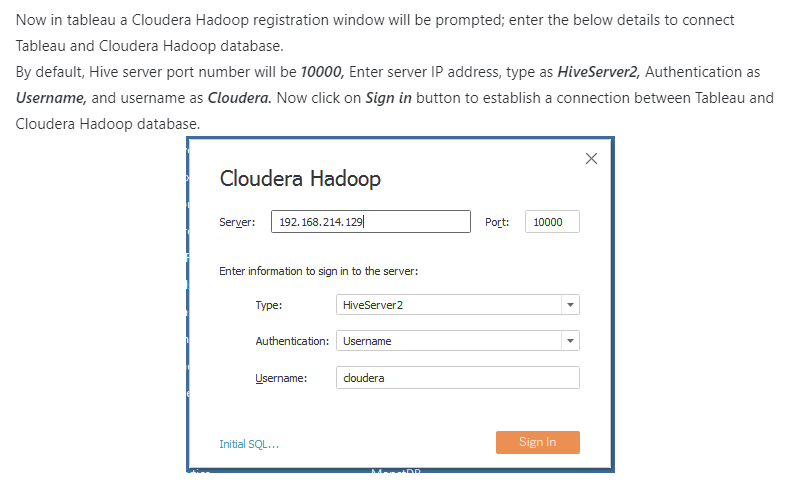
****

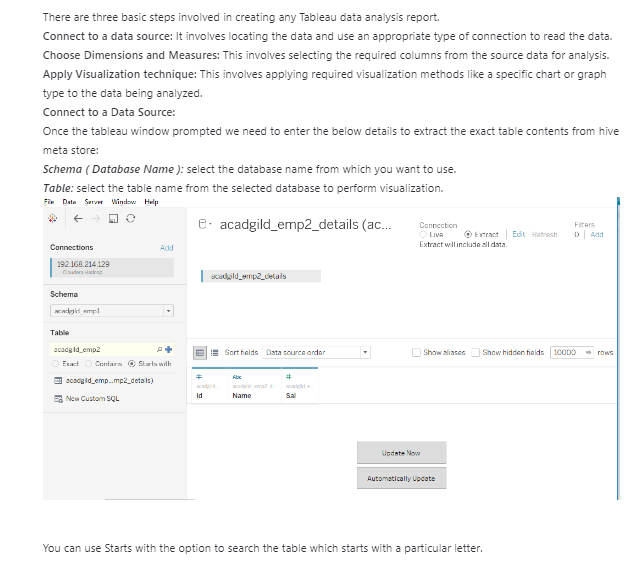
****

****

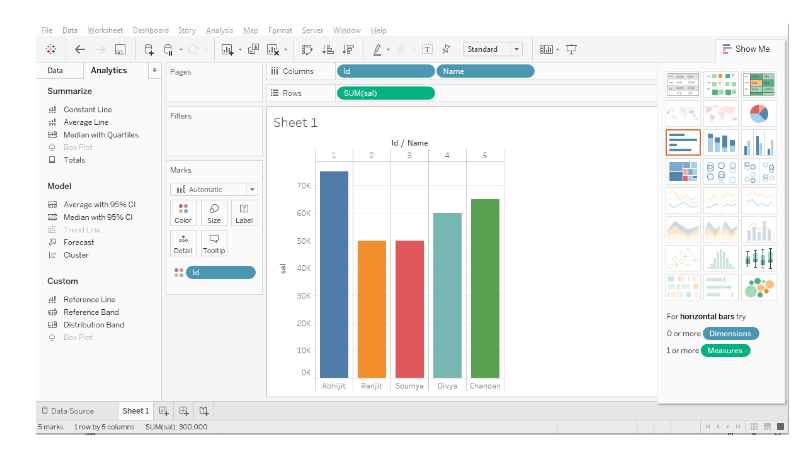
****

****

****

****

****

****

* **Conclusion**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Machine Type** | **AWS**  **Instance Name** | **Workload/**  **Cluster Type** | **Storage** | **CPU** | **Memory** | **Network** |
| **Worker Node**  **(Data Node)** | M4.2xlarge | Balanced workload | 4 Nodes Of 20 TB | 8 | 32 GB | 1 GB onboard, 2x10 GBE mezzanine/ external |
| C4.4xlarge | Compute Intensive workload | 4 Nodes Of 24 TB | 16 | 30 GB | 1 GB onboard, 2x10 GBE mezzanine/ external |
| D2.2xlarge | Storage Heavy workload | 4 Nodes Of 28 TB | 8 | 61 GB | 1 GB onboard, 2x10 GBE mezzanine/ external |
| **Master Node**  **(Namenode)** |  | Balanced workload | One Node Of 2TB RAID | 8 | 128 GB | 1 GB onboard, 2x10 GBE mezzanine/ external |
| **Secondary Namenode** |  | Balanced workload | One Node Of 2TB RAID | 8 | 128 GB | 1 GB onboard, 2x10 GBE mezzanine/ external |
| **Active Resource Manager** |  | Balanced workload | One Node Of 2TB RAID | 8 | 128 GB | 1 GB onboard, 2x10 GBE mezzanine/ external |
| **Standby Resource Manager** |  | Balanced workload | One Node Of 2TB RAID | 8 | 128 GB | 1 GB onboard, 2x10 GBE mezzanine/ external |

* **References**

1. <https://www.cloudera.com/documentation/enterprise/5-7-x/PDF/cloudera-administration.pdf>
2. <https://hadoop.apache.org/>