



# Installation steps on a cloud environment

Revision 20240429

## NOTE

This document is confidential and proprietary of **Denodo Technologies**.  
No part of this document may be reproduced in any form by any means without prior written authorization of **Denodo Technologies**.

Copyright © 2024  
Denodo Technologies Proprietary and Confidential

# 1

## 2 GOAL

---

This document describes the steps needed to install and configure the Denodo Platform on a Cloud environment like Amazon EC2, Azure Compute or Google Compute Engine.

## 3 CONTENT

---

### Server Side:

1. When creating a new VM instance in EC2 the only additional step you need to do is add a firewall rule. In the Configure Security Group step, click on 'Add Rule' and select 'Custom TCP Rule' in the 'Type' column. Enter the value '9995-9999' and '19995-19999' in the 'Port Range' text box and give a source IP. You can also specify another custom rule for the Denodo web container (9090) and Solutions Manager web container (19090):

Inbound rules <span style="float: right; border: 1px solid black; padding: 2px 5px;">Edit inbound rules</span>				
Type	Protocol	Port range	Source	Description - optional
Custom TCP	TCP	10090 - 10091	0.0.0.0/0	Solution Manager, License Manager
Custom TCP	TCP	9996	0.0.0.0/0	9996 for ODBC
Custom TCP	TCP	9999	0.0.0.0/0	9999 for JDBC
Custom TCP	TCP	19090	0.0.0.0/0	Solution Manager Web Port
Custom TCP	TCP	9995	0.0.0.0/0	JMX Auxillary
SSH	TCP	22	0.0.0.0/0	SSH Port
Custom TCP	TCP	9090	0.0.0.0/0	HTTP port
Custom TCP	TCP	9997	0.0.0.0/0	JMX Monitor Port

Similarly in Google Compute Engine, after creating a new instance add a new firewall rule for Denodo:

You can add the Google Cloud command that includes all ports :

```
gcloud compute --project=denodo4gcp-dev firewall-rules create denodo
```

```
--description="Denodo Platform 9.0 plus ssh" --direction=INGRESS  
--priority=1000 --network=default --action=ALLOW  
--rules=tcp:9995,tcp:9996,tcp:9997,tcp:9998,tcp:9999,tcp:9090,tcp:19995,t  
cp:19996,tcp:19997,tcp:19998,tcp:19999,tcp:19090 --source-  
ranges=172.16.0.0/25
```

The screenshot shows the Google Cloud Platform interface for creating a firewall rule. The left sidebar lists VPC network options, with 'Firewall' selected. The main content area is titled 'Create a firewall rule' and includes the following configuration details:

- Name:** denodo
- Description:** Denodo Platform 8.0 plus ssh
- Logs:** Off
- Network:** default
- Priority:** 1000
- Direction of traffic:** Ingress
- Action on match:** Allow
- Targets:** All instances in the network
- Source filter:** IP ranges
- Source IP ranges:** 172.16.0.0/25
- Second source filter:** None
- Protocols and ports:** Specified protocols and ports
  - tcp: 9995,9996,9997,9998,9999,9090,19995,19996,19997,19998,19999
  - udp: all
  - Other protocols

In Denodo 9.0 and Denodo 8.0, the JDBC driver, the administration tool and the new Design Studio [connections now use only one port](#) to communicate with Virtual DataPort

(by default, 9999). In previous versions, the communication requires two ports (9999 and 9997). This change simplifies the configuration of firewalls.

2. Perform a headless installation of the Denodo Platform (See the Knowledge Base article [Headless installation](#)).

3. Next, you have to add an entry to the hosts file. First, get the local IP address with the below command:

```
$ ifconfig | grep 'inet addr: '| grep -v '127.0.0.1'
```

You will get an output like below, showing the internal IP address of the server:

```
inet addr:10.240.84.182 Bcast:10.240.84.182 Mask:255.255.255.255
```

4. Now you need to add an entry to the hosts file with the match between this IP address and the hostname that will be used for the connections from the clients. This hostname can be either resolvable by the client's DNS (a public hostname) or a private one. To configure the hosts file follow these steps:

- a. Edit the `/etc/hosts` file in the Cloud server and add an entry with the private IP and the hostname.

*Example of entry with public hostname:*

```
10.240.84.182 ec2-35-35-252-633-compute-1.amazonaws.com
```

*Example of entry with private hostname:*

```
10.240.84.182 denodovm
```

- b. Add the same value of hostname to the `registryURL` parameter of the `conf/vdp/VDBConfiguration.properties` file so it will be look like:

```
com.denodo.vdb.vdbinterface.server.VDBManagerImpl.registryURL  
=<hostname>
```

(replace `<hostname>` with the hostname used in the previous step)

5. Finally, regenerate the startup scripts and start the VDP server:

```
$(DENODO_HOME)/bin/regenerateFiles.sh  
$(DENODO_HOME)/bin/vq1server_startup.sh
```

(replace `<DENODO_HOME>` with the path configured during installation)

## Client Side:

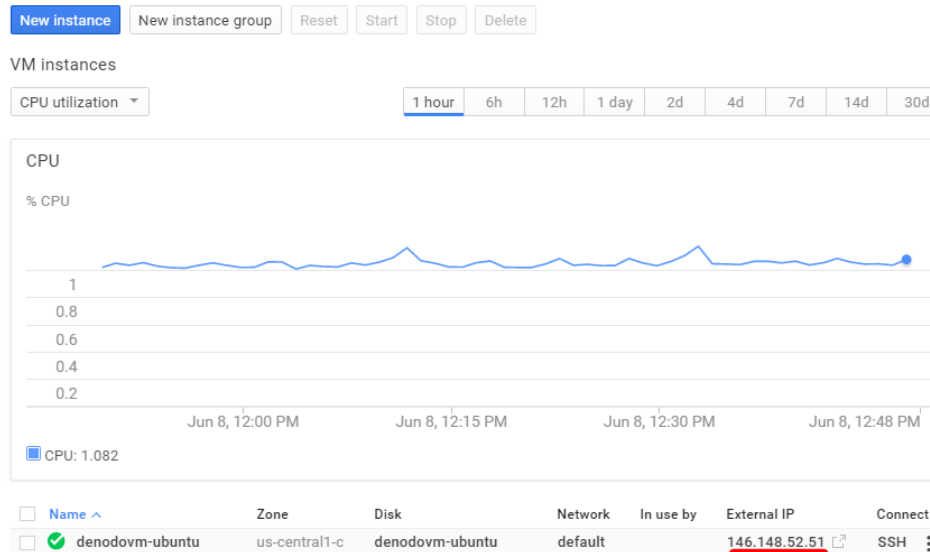
In order for clients to be able to connect to this Denodo running in the Cloud server, they need to be able to resolve the hostname used during the configuration of the server.

If you have used a hostname that can be resolved by the client's DNS (a public hostname) you do not need to do any additional configuration. However, if the

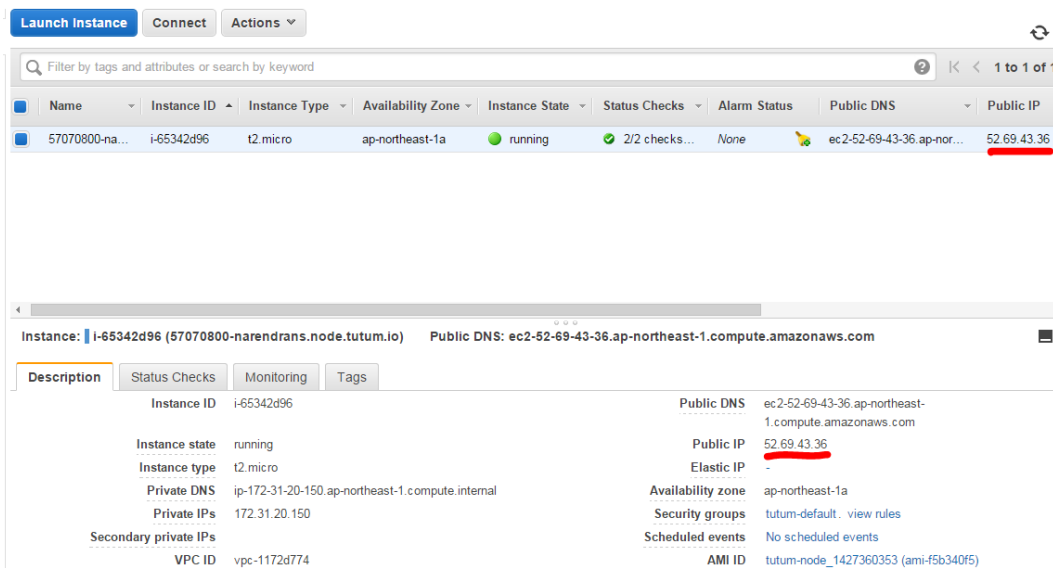
hostname cannot be resolved by the clients, you need to add it to the hosts file of the clients.

As an example, if you are using a private hostname like denodovm, you can follow the next steps:

1. On the client side, you have to add an entry in the hosts file with the public IP of the instance. First, you need to identify the public IP address of your Cloud server. The procedure depends on the Cloud provider that you are using:



## Finding the public IP in Google Compute Engine



## Finding the public IP in Amazon EC2

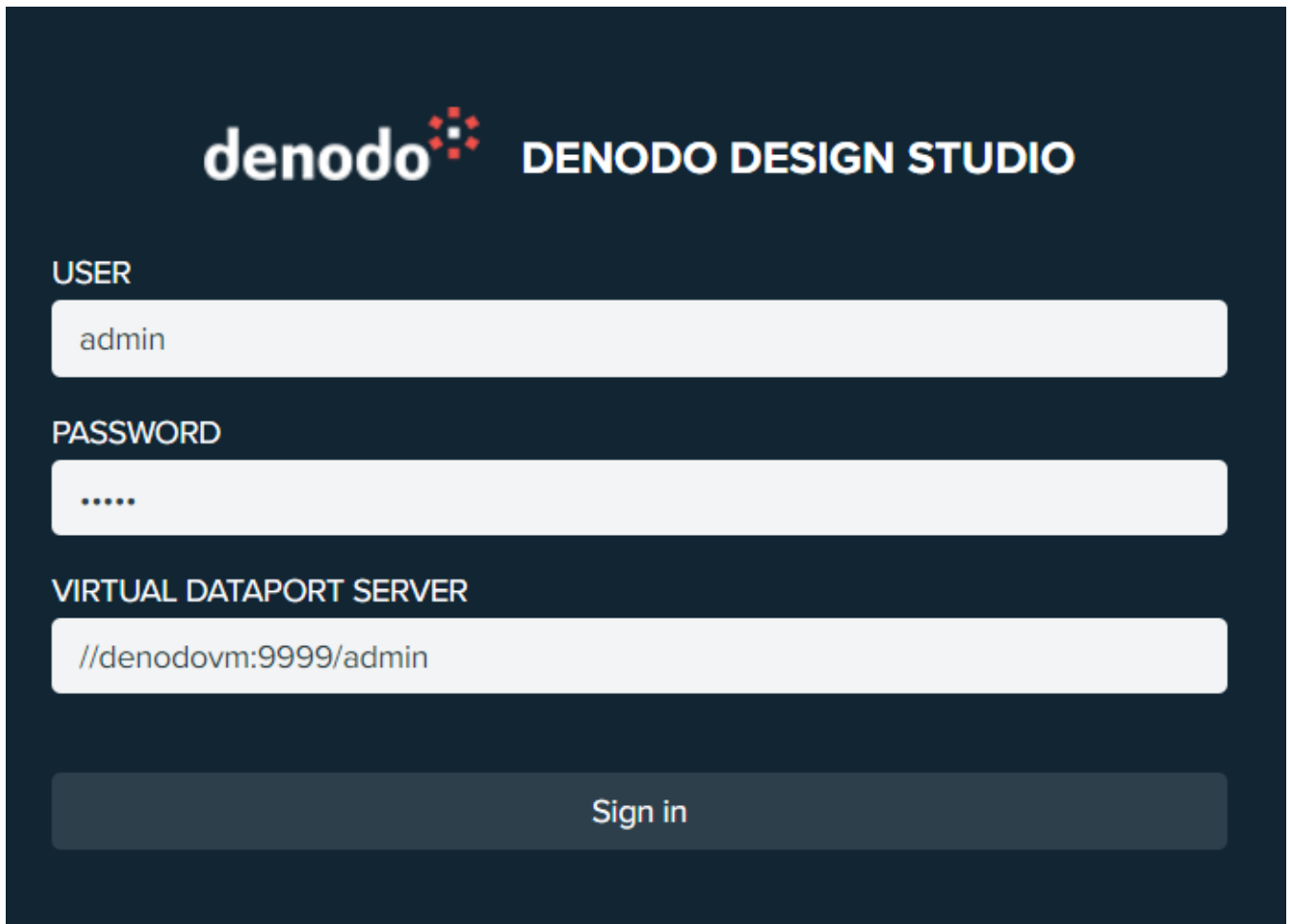
2. Then you need to add this public IP address to the hosts file. For example, if

the public IP is 146.148.52.51, you can add an entry like the one below:

```
146.148.52.51 denodovm
```

(the hosts file location in Linux is /etc/hosts and in Windows:  
%SYSTEMROOT%\System32\Drivers\etc\hosts)

3. You can now connect to the VDP server using the Design Studio.



**denodo** DENODO DESIGN STUDIO

USER  
admin

PASSWORD  
\*\*\*\*\*

VIRTUAL DATAPORT SERVER  
//denodovm:9999/admin

Sign in

## 4 REFERENCES

---

[Headless installation](#)

[Denodo Platform Ports](#)

[Solution Manager Ports](#)

Google Compute Engine: <https://cloud.google.com/compute/>

Amazon EC2: <http://aws.amazon.com/ec2/>

Azure Compute: <https://azure.microsoft.com/en-us/services/virtual-machines/>