Cloud Authentication Lab

# CCNP Lab 5

## Axel Li

## 12 December 2018

# Purpose

The purpose of this lab was to practice implementing a Linux TACACS+ server and a Windows RADIUS server on for each of AWS and Azure, commonly used cloud services. For each combination of the cloud service and protocol, we were to configure a server on a virtual machine in the cloud and configure a physical router as the corresponding client. We then had to confirm that authentication was functioning correctly with only authenticated users able to access the routers.

# Background

Cloud computing has become more prevalent in the recent years due to the numerous benefits it provides. By using cloud computing, companies can minimize or avoid initial IT costs, set up applications faster, improve manageability, and reduce maintenance needs under a “pay-as-you-go” model. Compared to traditional servers, servers on the cloud can be easily set up, scaled, or upgraded without requiring hardware and software installations. The flexibility and wide array of services offered on the cloud also make cloud computing incredibly popular. Currently, cloud computing is used in a variety of industries, ranging from manufacturing and commerce to health and science. Therefore, it is important to have experience and skills in working in the cloud.

# Summary

Before configuring any of the routers, my partner and I registered educational accounts from the local community college to receive student subscriptions to Azure and AWS. We then proceeded to create the virtual machines and set up servers on the VMs, using RDP to configure the Windows RADIUS server and SSH (or the Azure serial console) to configure the Ubuntu TACACS+ server.

Like before, we installed the TACACS+ daemon and edited the configuration files manually to add the required users and user groups, then started the service to run on the Ubuntu virtual machine. For the RADIUS server, we promoted the active directory domain service to a new forest, configured a connection request policy, configured a RADIUS client and shared key, created a new user group and added a configured user, and configured a network policy. For this lab, the IP addresses of the servers were automatically assigned by the respective cloud service.

In order to get internet access to the router, we used bridging through the desktop computer it was physically connected to, which enabled the router to connect to the internet via the Wi-Fi adapter of the desktop. In a saved file containing the configuration of the routers, we set up the necessary TACACS+, RADIUS, and AAA commands, then pasted the configuration into the routers to complete the setup for the router. For this lab, the routers used DHCP addresses instead of statically configured addresses.

To test the devices, we attempted to log in via console to the routers to test that the authentication was correct, and verified that authentication was working properly with Wireshark, a packet sniffing program, and the records available in the Network Policy and Access Services for RADIUS.

# Commands

The key commands used in this lab for the routers were:

aaa new-model - enables new access control commands and functions

aaa authentication login default group tacacs+ local line none – configures AAA authentication via TACACS+ at login (defaults to the local database, then line password if unavailable)

aaa authentication enable default group tacacs+ enable none – configures AAA authentication via TACACS+ for when users attempt to access the privileged command level (defaults to the enable password if unavailable)

login authentication default - enables AAA authentication for logins

username [username] privilege 15 secret [password] - create a local username and password for access to router with privilege 15 (for debugging and as a backup)

ip address dhcp – configures an interface of the router to use an DHCP address

tacacs server [tacacs server configuration name] – enters TACACS+ server configuration mode

address ipv4 172.16.1.2 – specifies the IP address of the TACACS+ server

key [shared key] – specifies the shared key between the TACACS+ server and client

aaa group server tacacs+ [server-group name] – defines an AAA TACACS+ server group with the specified group name

server name [tacacs server name] – adds the TACACS+ server with the specified name to the server group

ip tacacs source-int g0/0 - specifies the interface for TACACS+

test aaa group tacacs+ [username] [password] new-code - attempts to log in via TACACS+ with the specified username and password

aaa authentication login default group radius local – configures AAA authentication via RADIUS at login (defaults to the local database if unavailable)

aaa authorization exec default group radius if-authenticated - configures AAA authentication via RADIUS for when users attempt to access the privileged command level (does not request if already authenticated)

radius-server host [server IP address] key [shared-key] - specifies the RADIUS host by IP address and the authentication/encryption key shared between the client and the RADIUS server

The key commands used in this lab for the TACACS+ server running on Ubuntu were:

service tacacs\_plus start – starts the TACACS+ daemon

tac\_pwd – produces a DES encryption of a specified password

systemctl status tacacs\_plus.service – checks the status of the TACACS+ daemon

nano – edits the contents of a specified file

# Tables and Diagrams

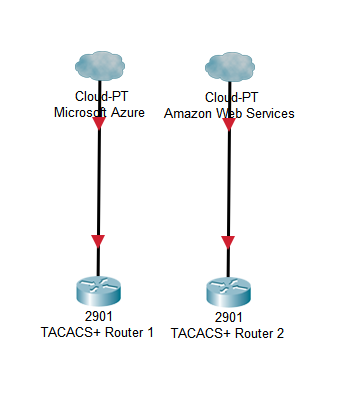
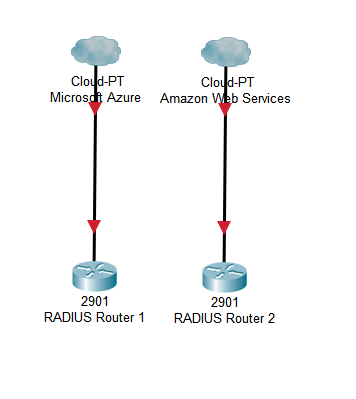
TACACS+ Configurations:

|  |  |  |
| --- | --- | --- |
| user | “tacuser” | “netuser” |
| login | “taclogin” | “netlogin” |
| enable | “tacenable” | “netenable” |

RADIUS Configurations:

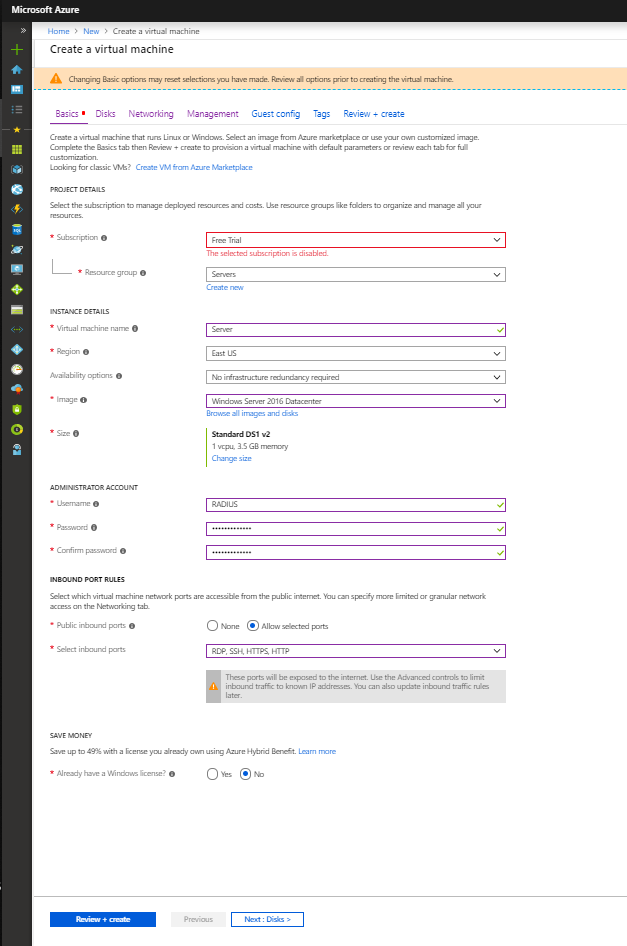
|  |  |
| --- | --- |
| user | “bob” |
| login | “qwer1234Q” |

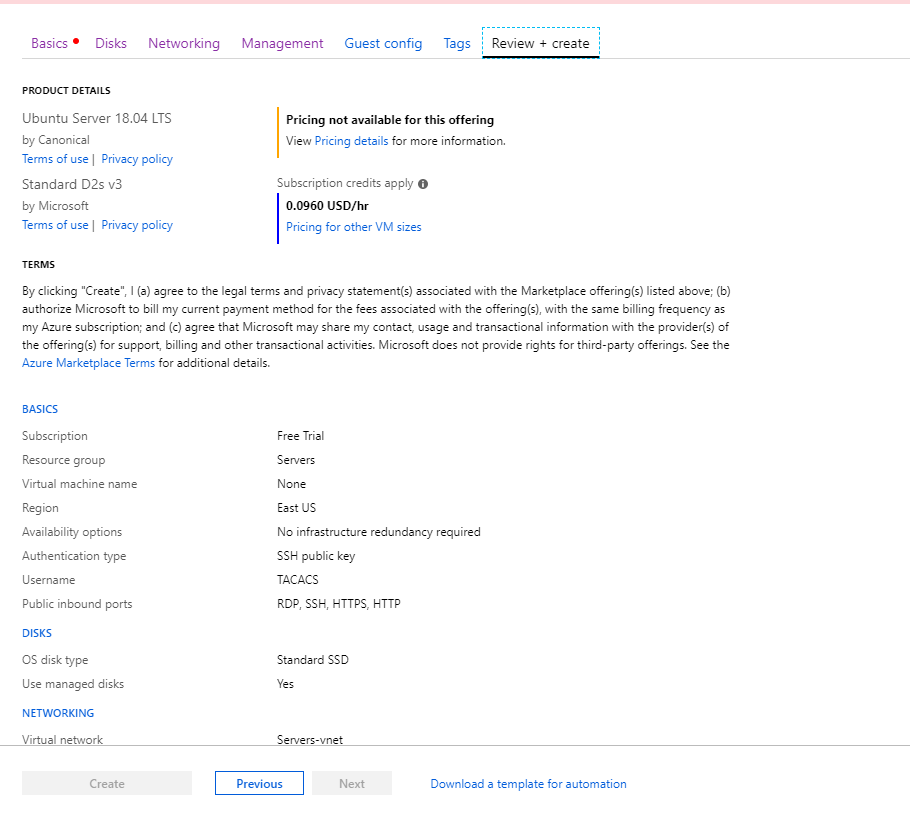
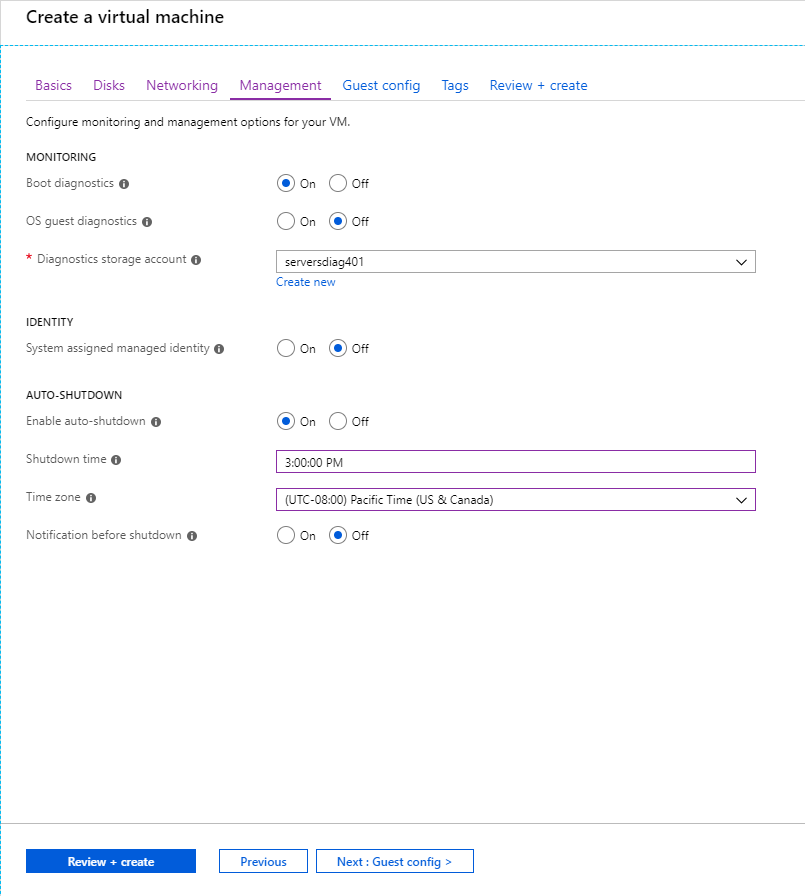
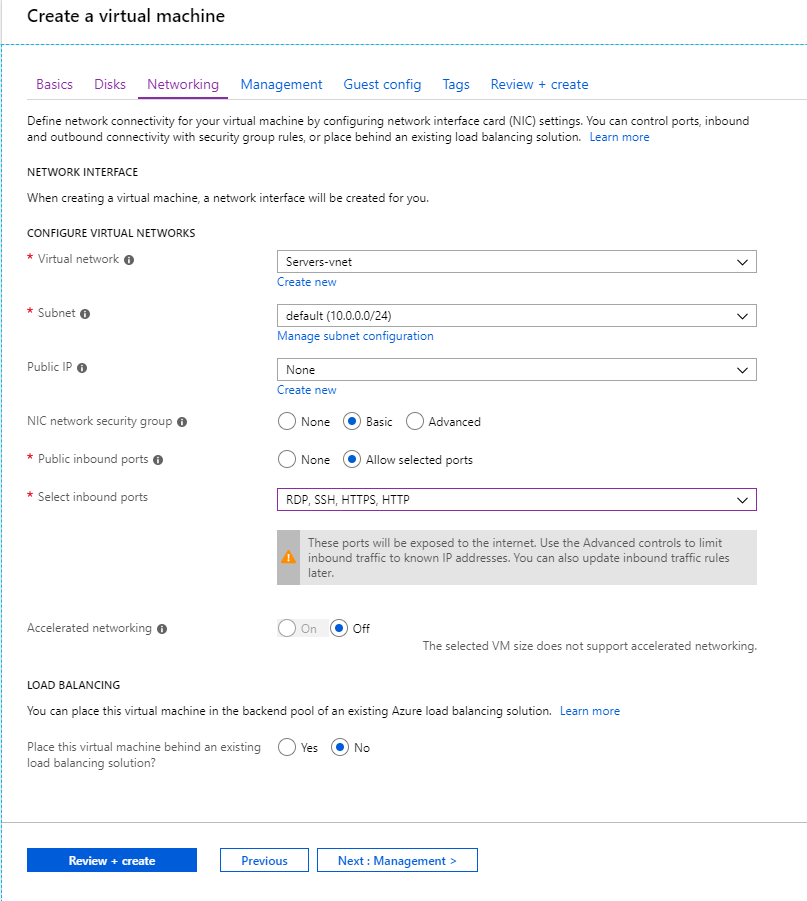
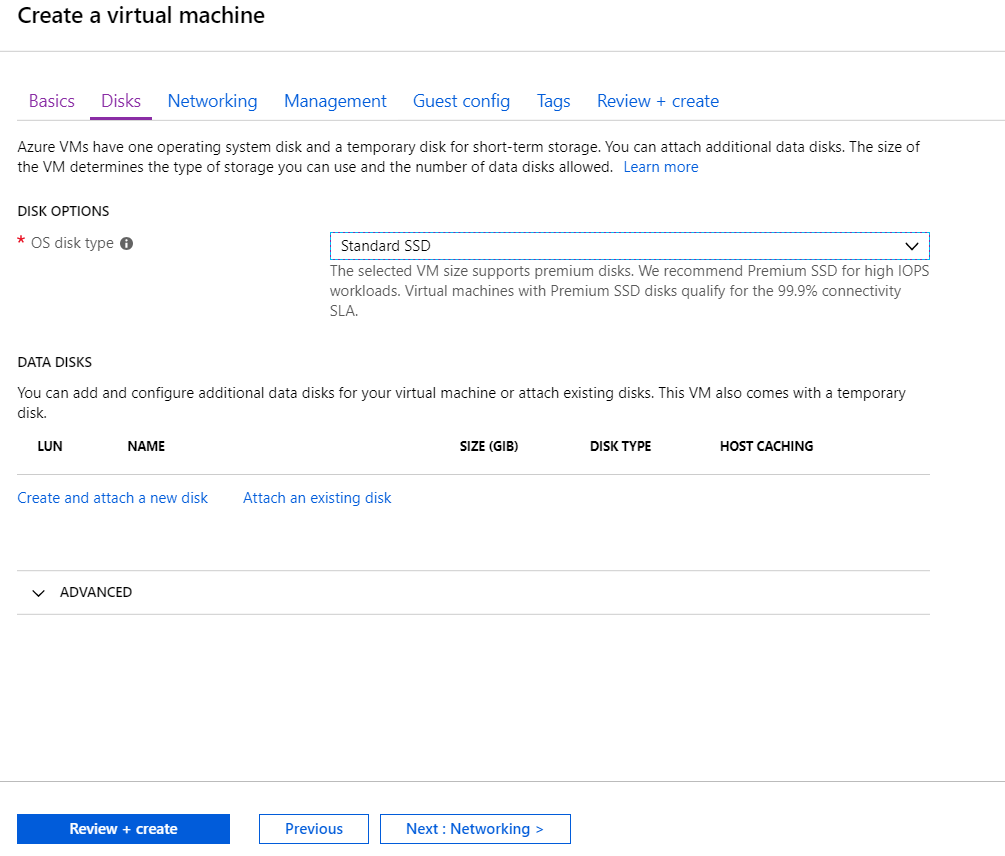
Topology:



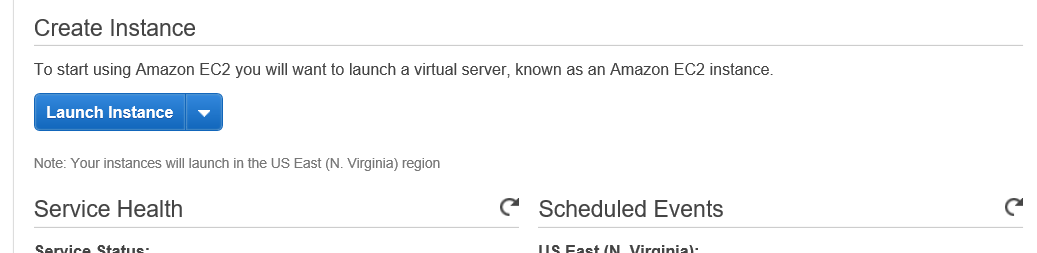
# Configurations

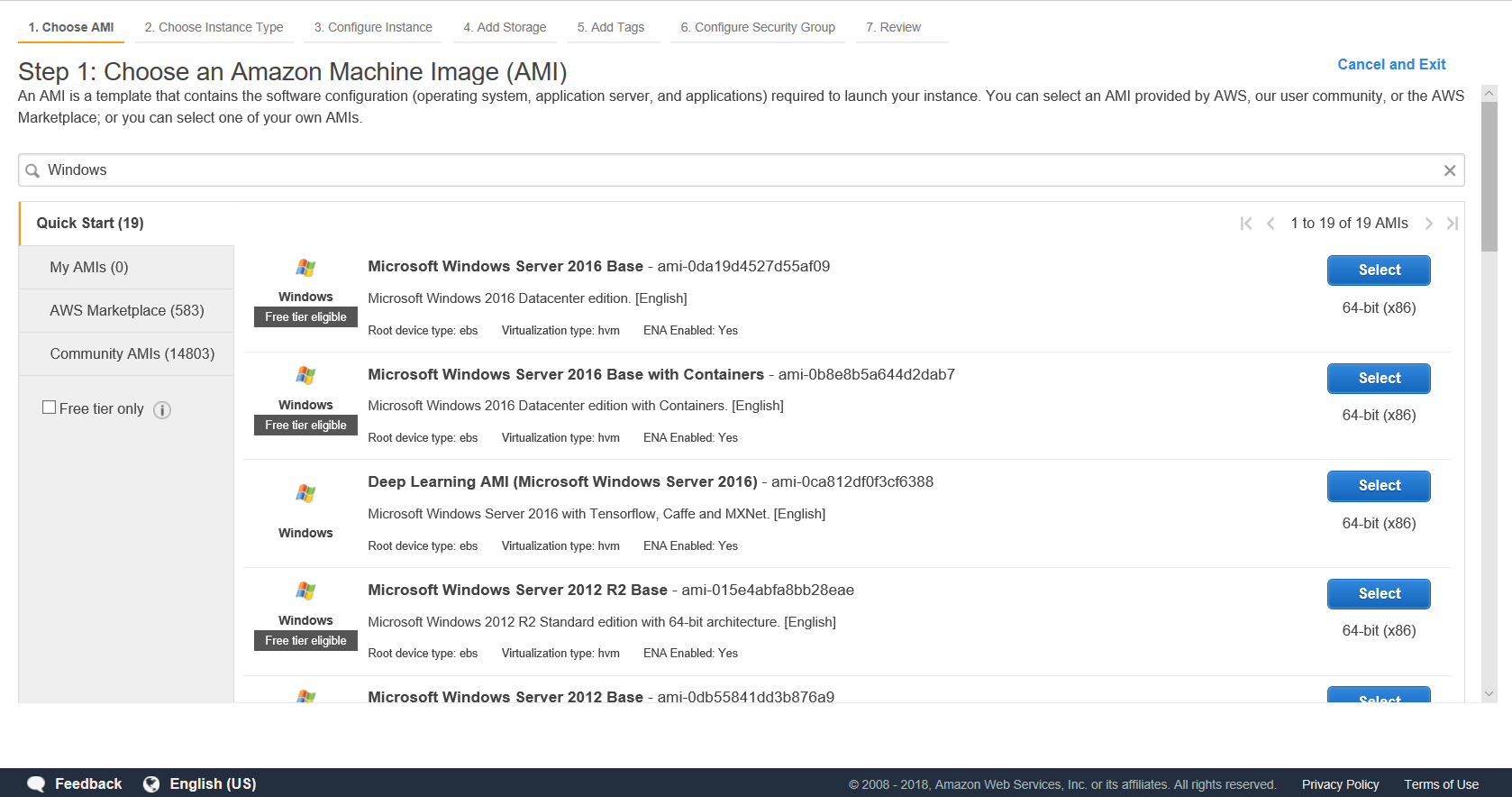
Azure VM Setup:

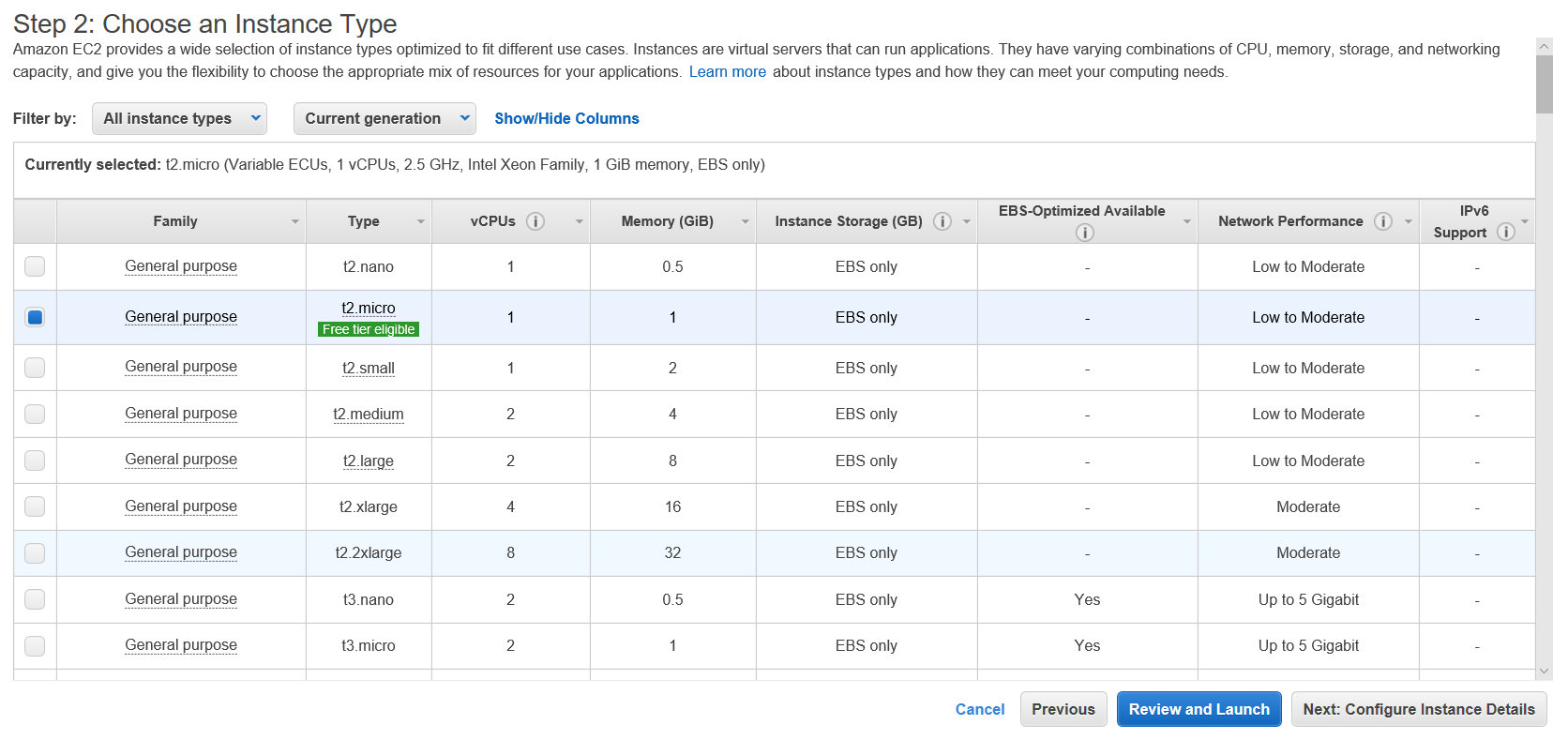


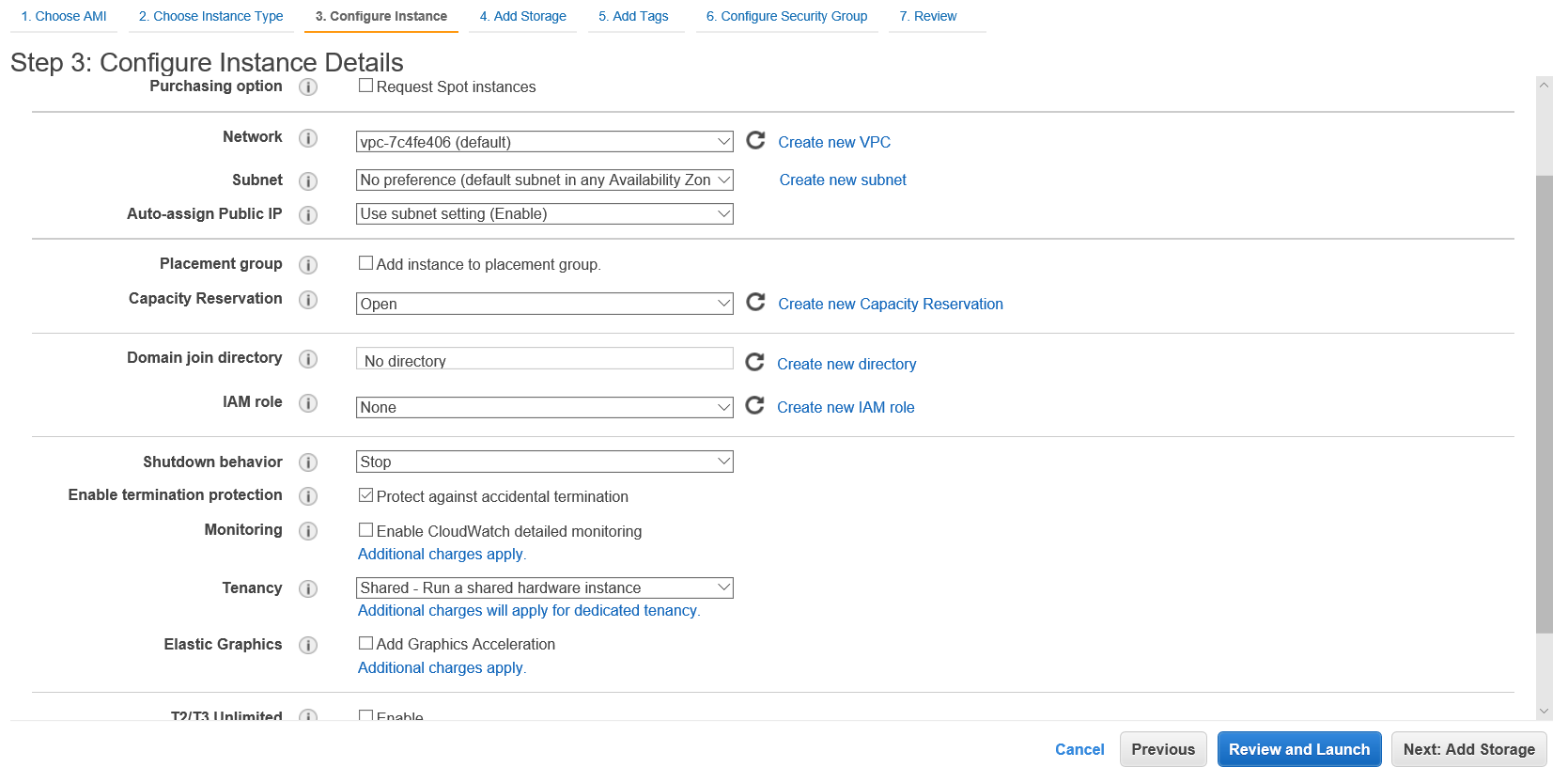


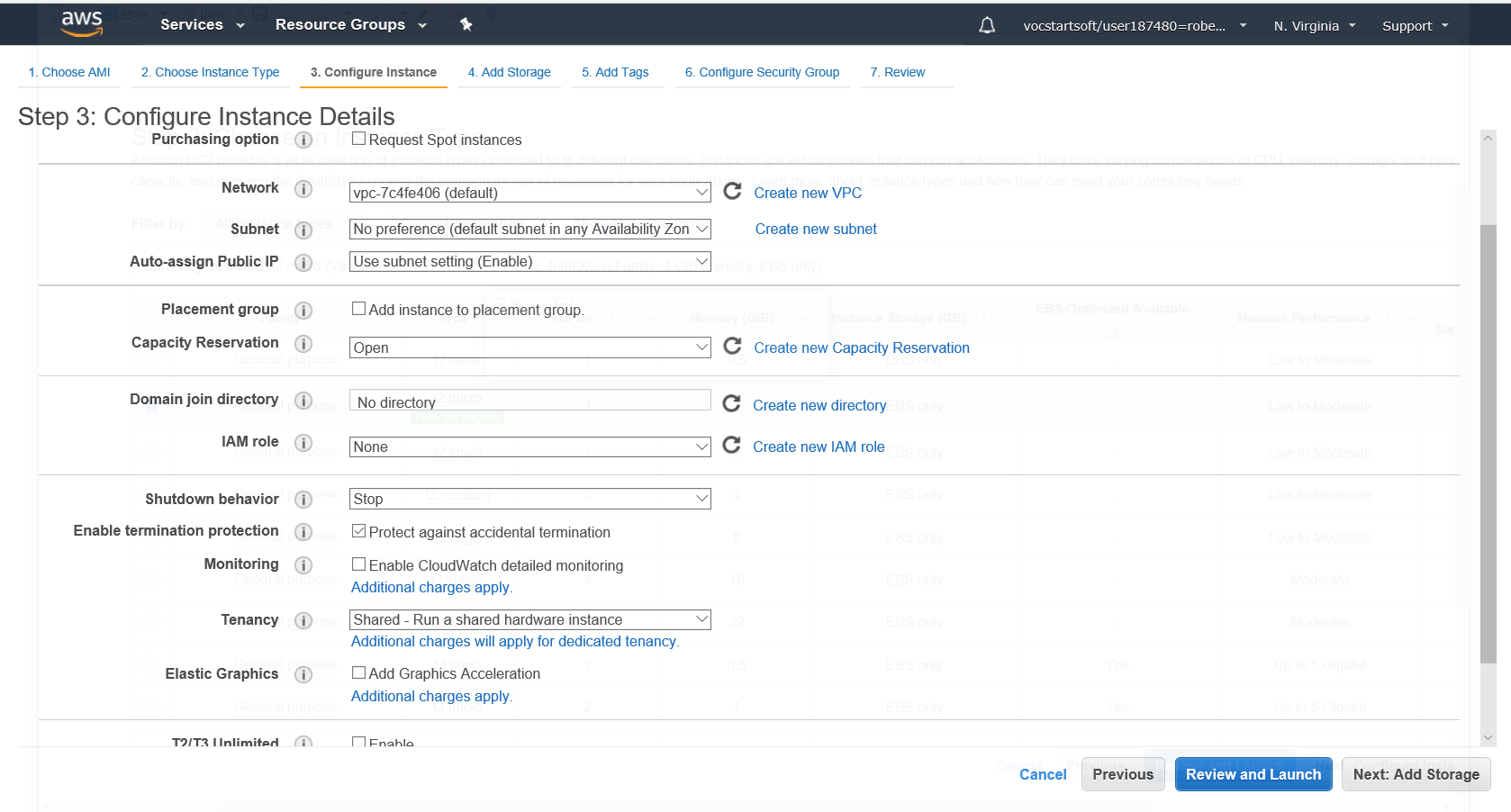
AWS VM Setup:

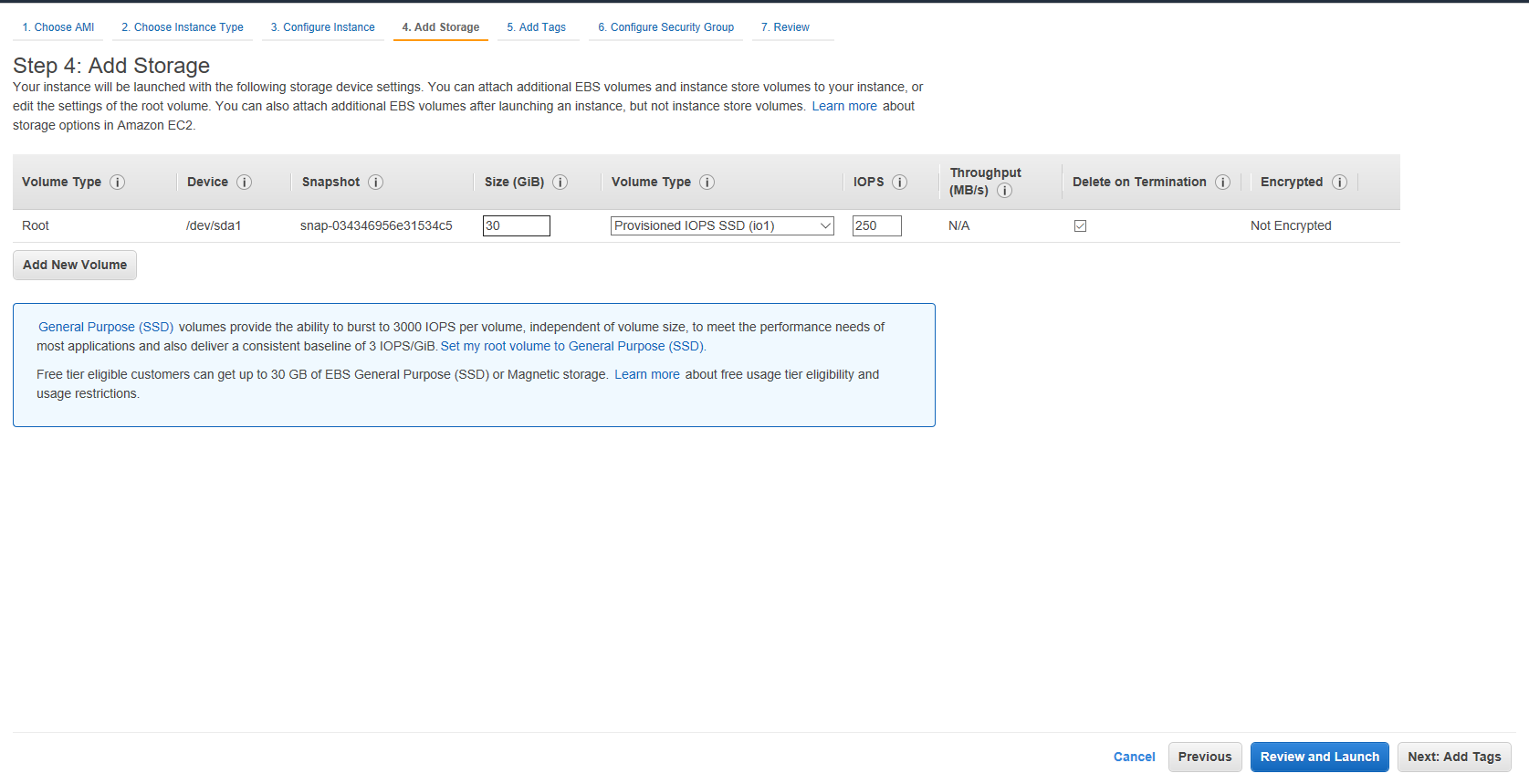


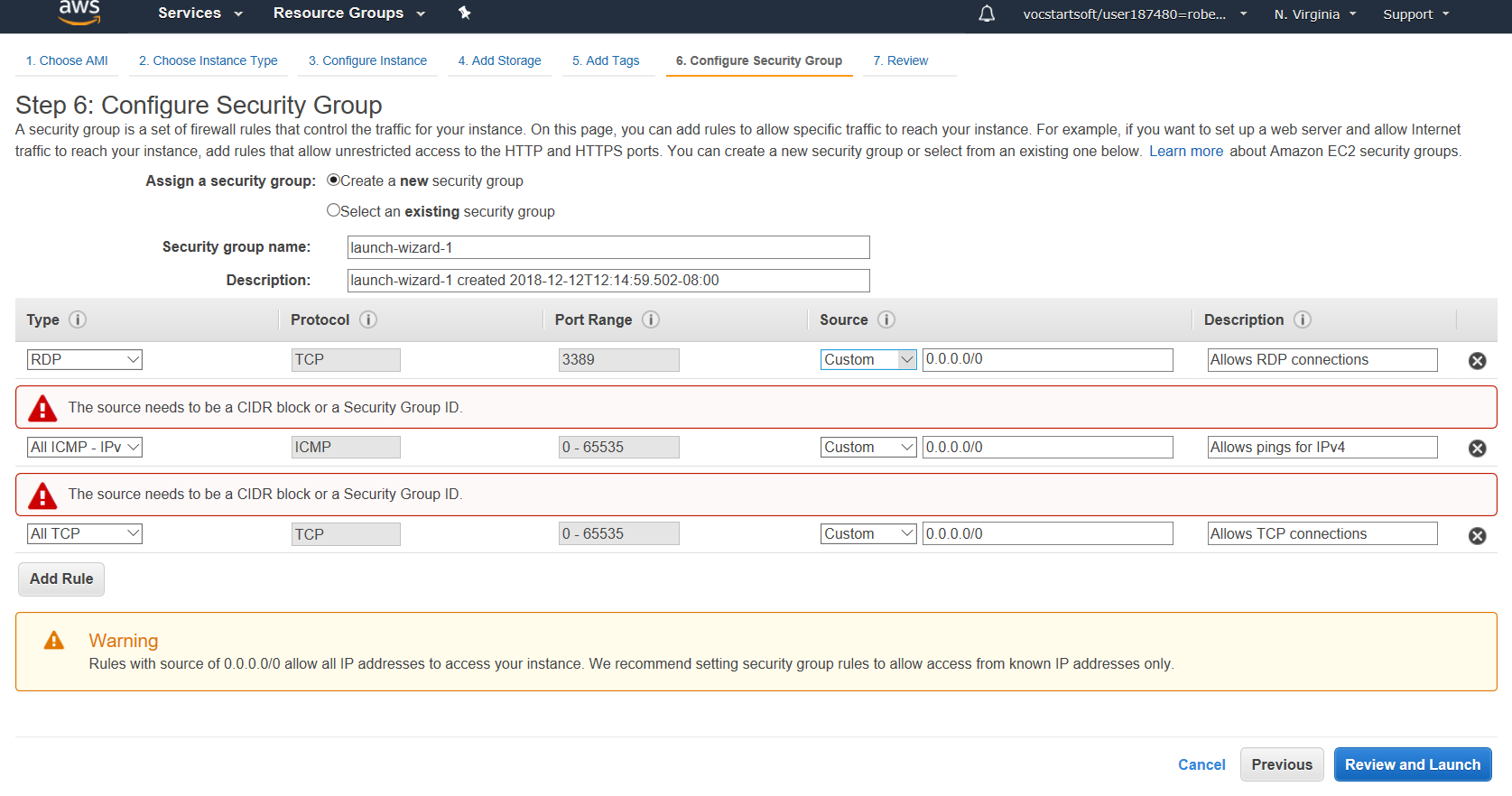


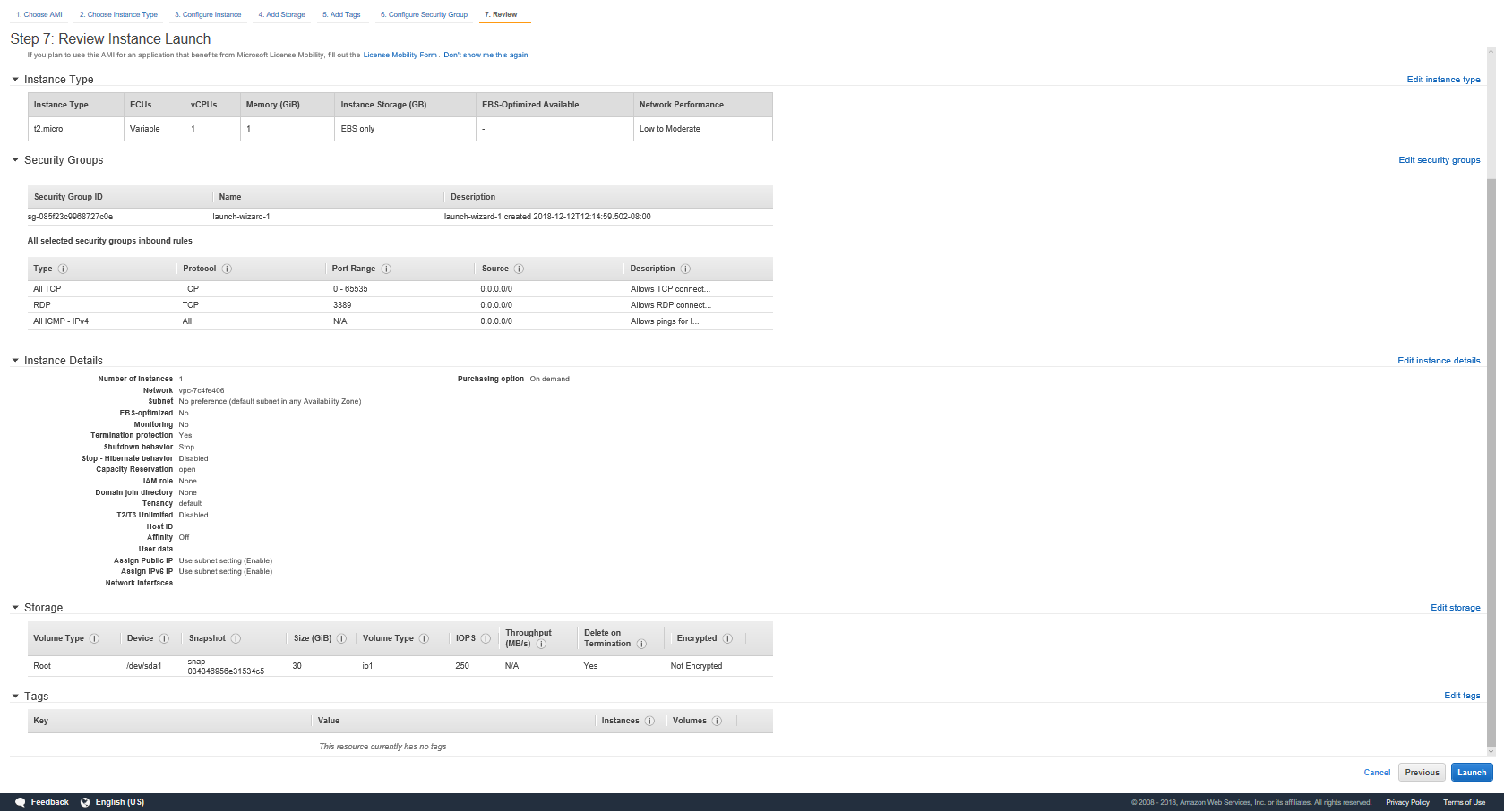












**TACACS+ Client:**

**R1#** show run

service password-encryption

hostname R1

enable secret cisco

aaa new-model

aaa group server tacacs+ tacser

aaa authentication login default group tacacs+ local line none

aaa authentication enable default group tacacs+ enable none

aaa accounting exec default start-stop group tacacs+

aaa accounting commands 0 default start-stop group tacacs+

aaa accounting commands 15 default start-stop group tacacs+

interface GigabitEthernet0/0

ip address dhcp

no shutdown

ip tacacs source-interface GigabitEthernet0/0

tacacs-server host 18.207.144.230

tacacs-server key 0 tackey

line con 0

password cisco

line vty 0 4

transport input all

line vty 5 15

transport input all

username backup privilege 15 password backup

**R1#show tacacs**

Tacacs+ Server -  public :

              Server name: tacauth

           Server address: 172.16.1.2

              Server port: 49

             Socket opens:       103

            Socket closes:        103

            Socket aborts:          0

            Socket errors:          0

          Socket Timeouts:          0

  Failed Connect Attempts:          3

       Total Packets Sent:        129

       Total Packets Recv:        129

**R1#show aaa sessions**

Total sessions since last reload: 19

Session Id: 40

  Unique Id: 30

  User Name: tacuser

  IP Address: 0.0.0.0

  Idle Time: 0

  CT Call Handle: 0

Session Id: 47

  Unique Id: 32

  User Name: netuser

  IP Address: 172.16.1.10

  Idle Time: 0

  CT Call Handle: 0

Ubuntu TACACS+ Server:

**/etc/tacacs+/tac\_plus.conf**

accounting file = /var/log/tac\_plus.acct

key = tackey

group = network {

default service = permit

cmd = configure {

deny .\*

}

cmd = show {

deny running-config

}

}

user = netuser {

member = network

login = des 0DkzB4EbGjgBw

enable = des oNfJnxCB/lqS.

}

group = admin {

default service = permit

}

user = tacuser {

member = admin

login = des BIOVLNkpdP7PI

enable = des QMxu0r.4.3cDU

}

**/etc/default/tacacs+**

DAEMON\_OPTS="-C /etc/tacacs+/tac\_plus.conf"

RADIUS Client:

**RADIUS-AWS-CLIENT#**show run

Building configuration...

Current configuration : 2067 bytes

Last configuration change at 19:42:59 UTC Wed Dec 12 2018

version 15.2

service timestamps debug datetime msec

service timestamps log datetime msec

service password-encryption

hostname RADIUS-AWS-CLIENT

enable secret 4 tnhtc92DXBhelxjYk8LWJrPV36S2i4ntXrpb4RFmfqY

aaa new-model

aaa group server radius radser

server name radauth

aaa authentication login default group radius local line

aaa authentication enable default group radius enable

username backup privilege 15 secret 4 JB.BVbRMnjUKj2O3xMsmStGL2f/yjFT1G3wzz7uTIHA

interface GigabitEthernet0/0

ip address dhcp

duplex auto

speed auto

ip radius source-interface GigabitEthernet0/0

radius server radauth

address ipv4 34.204.43.234 auth-port 1645 acct-port 1646

key 7 095F4B0A0B0003

line con 0

password 7 094F42080A16

line aux 0

line 2

no activation-character

no exec

transport preferred none

transport output lat pad telnet rlogin lapb-ta mop udptn v120 ssh

stopbits 1

line vty 0 4

password 7 060503205F5D

transport input all

line vty 5 15

password 7 060503205F5D

transport input all

!

scheduler allocate 20000 1000

!

end

**RADIUS-AWS-CLIENT#**show dhcp lease

Temp IP addr: 192.168.40.25  for peer on Interface: GigabitEthernet0/0

Temp  sub net mask: 255.255.254.0

  DHCP Lease server: 192.168.40.1, state: 4 Requesting

  DHCP transaction id: 2555

  Lease: 1156 secs,  Renewal: 0 secs, Rebind: 0 secs

  Next timer fires after: 00:00:03

  Retry count: 1   Client-ID: cisco-24e9.b33c.1c60-Gi0/0

  Client-ID hex dump: 636973636F2D323465392E623333632E

                      316336302D4769302F30

  Hostname: RADIUS-AWS-CLIENT

**RADIUS-AWS-CLIENT#**show radius statistics

                                 Auth. Acct. Both

        Maximum inQ length:         NA NA 1

      Maximum waitQ length:         NA NA 1

      Maximum doneQ length:         NA NA 1

      Total responses seen:          5 0 5

    Packets with responses:          5 0 5

 Packets without responses:          1 0 1

 Access Rejects           : 3

Average response delay(ms):        100 0 100

Maximum response delay(ms):        140 0 140

 Number of Radius timeouts:          4 0 4

      Duplicate ID detects:          0 0 0

Buffer Allocation Failures:          0 0 0

Maximum Buffer Size (bytes):        101 0 101

Malformed Responses        : 0 0      0

Bad Authenticators         : 0 0      0

Unknown Responses          : 0 0      0

Source Port Range: (2 ports only)

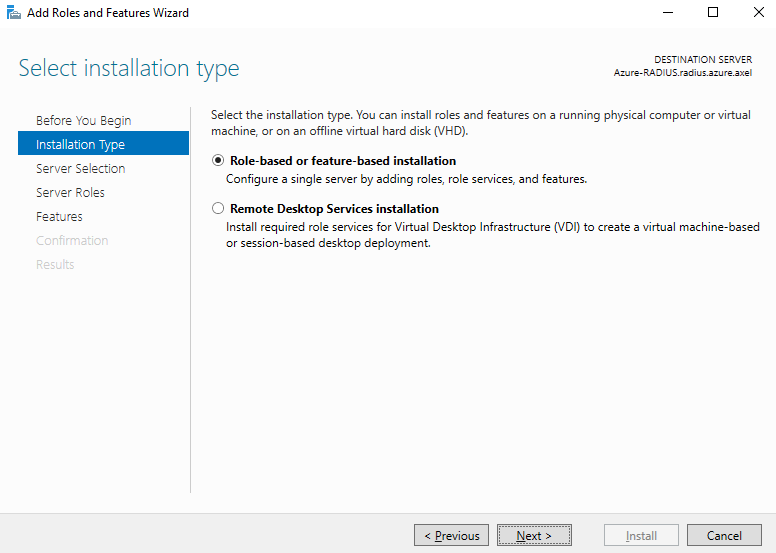
1645 - 1646

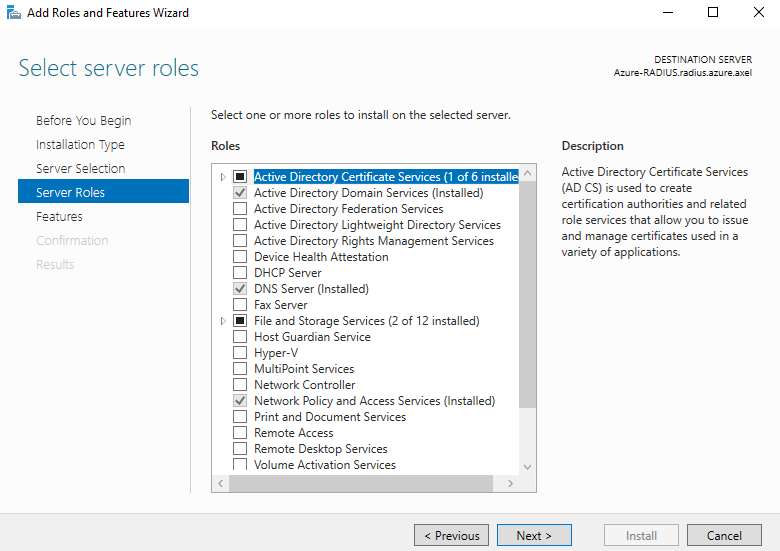
Last used Source Port/Identifier:

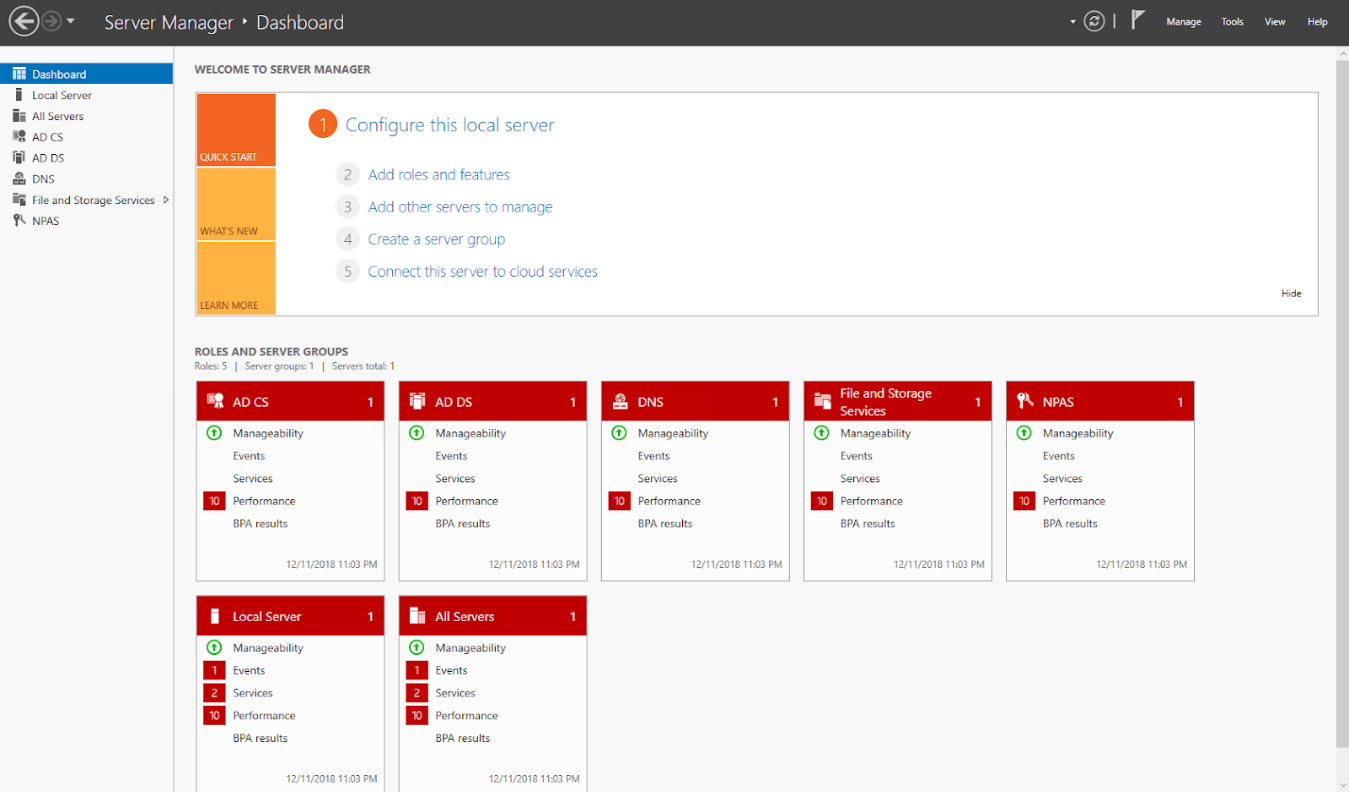
1645/6

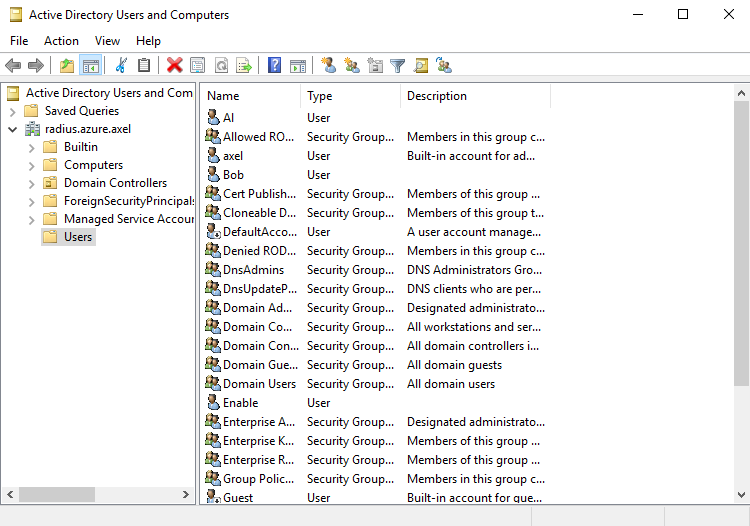
1646/0

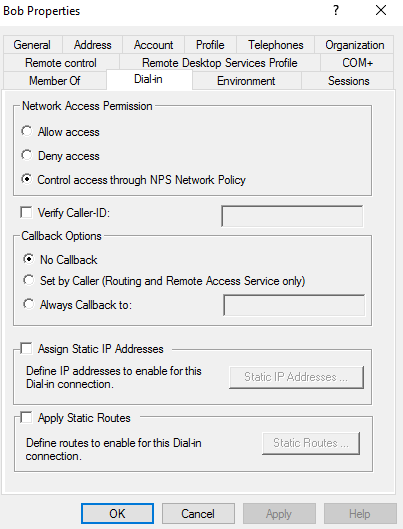
Windows TACACS+ Server:

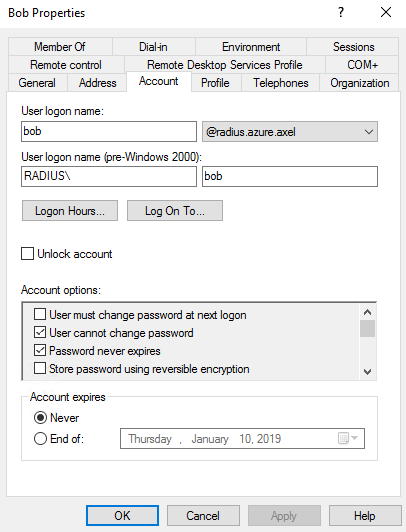
****

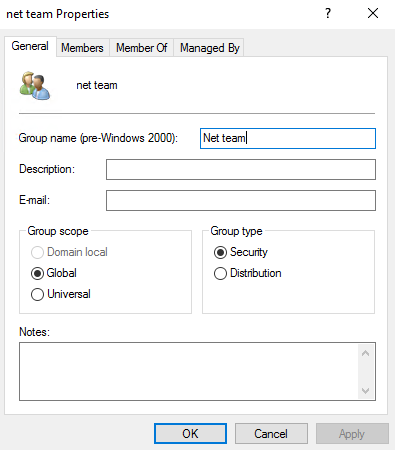
****

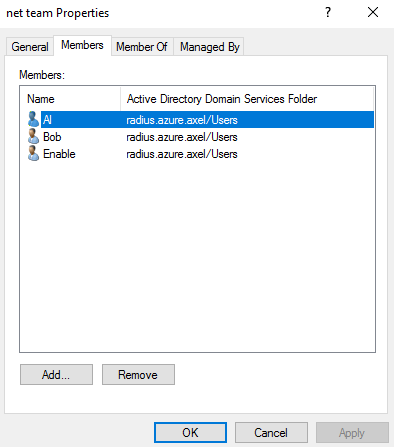
****

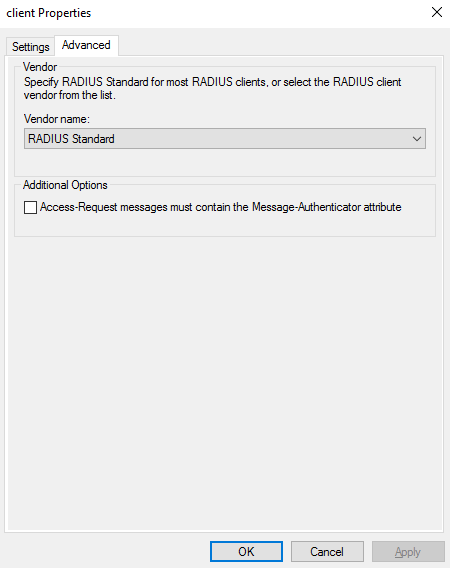
****

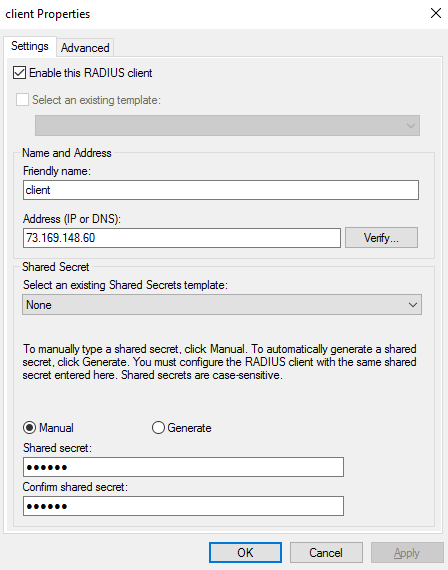
****

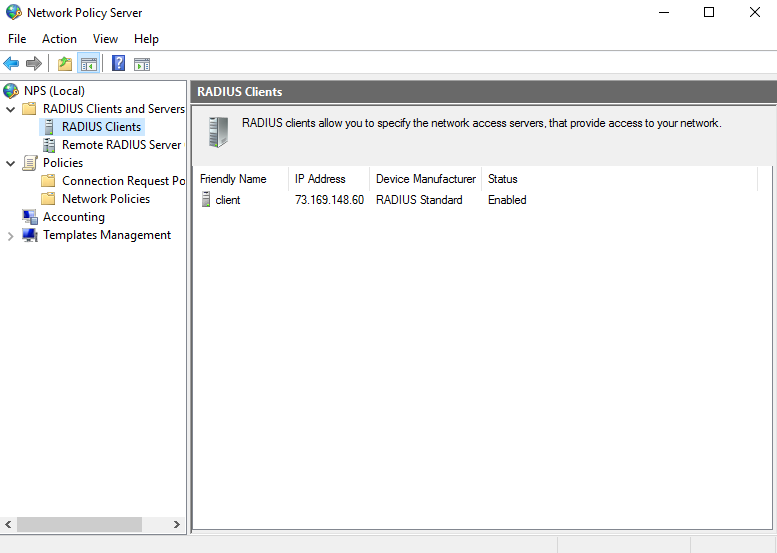


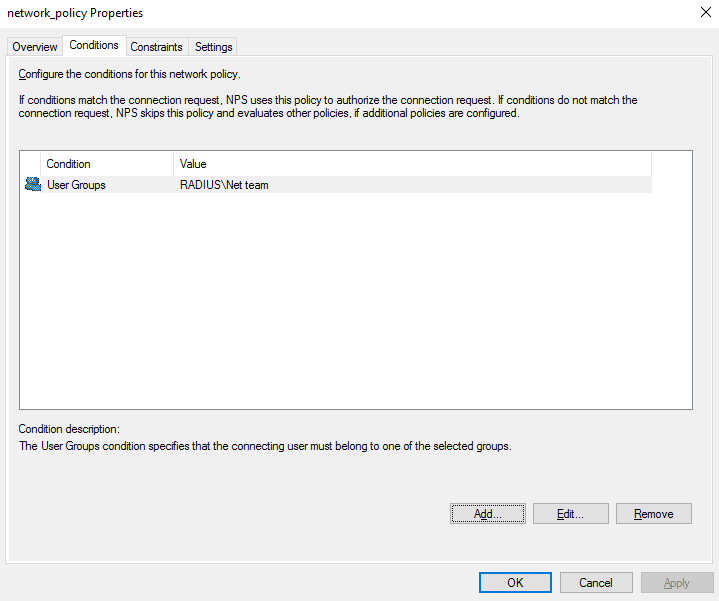
****

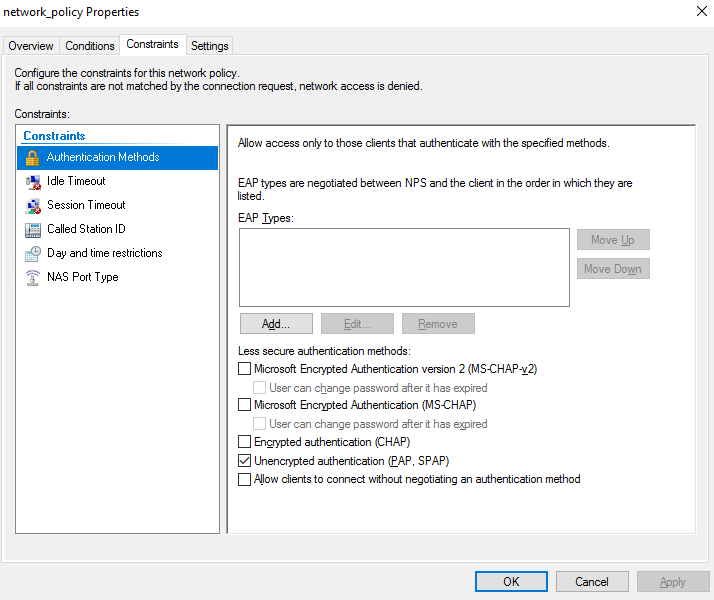
****

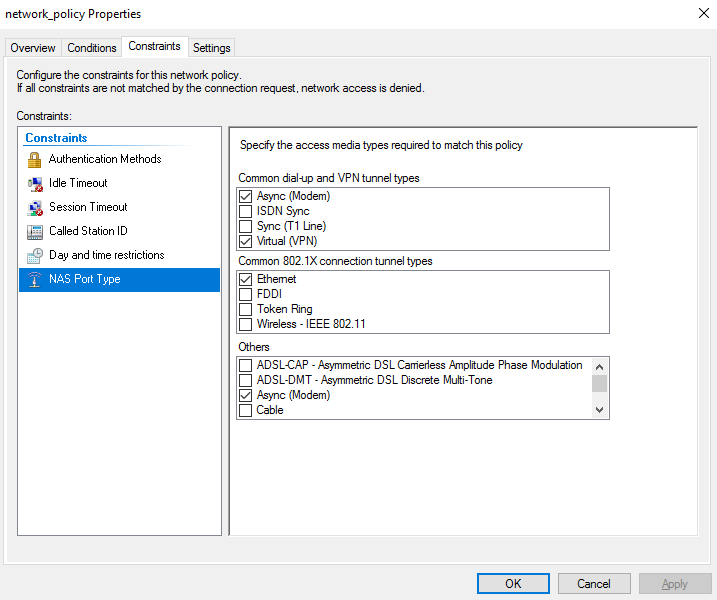
****

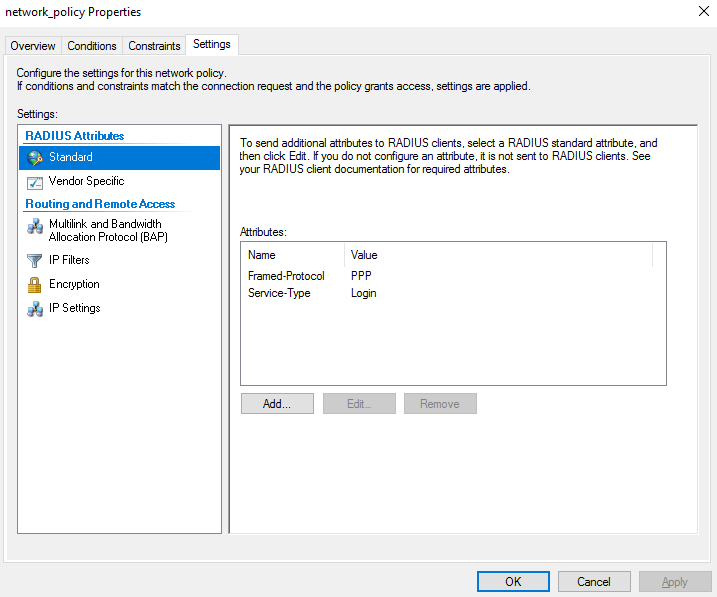
****

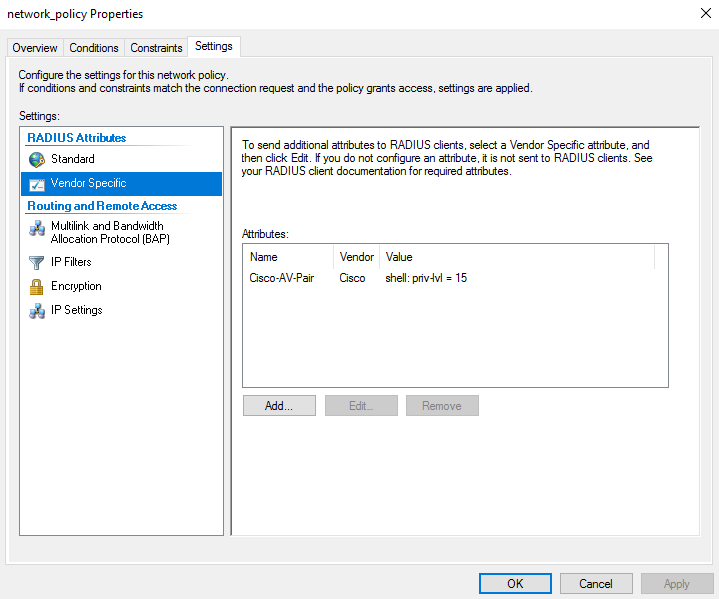
****

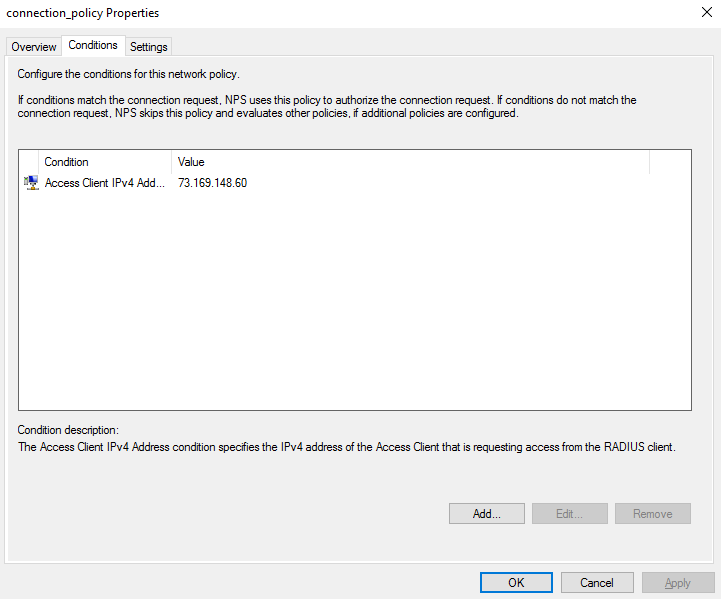
****

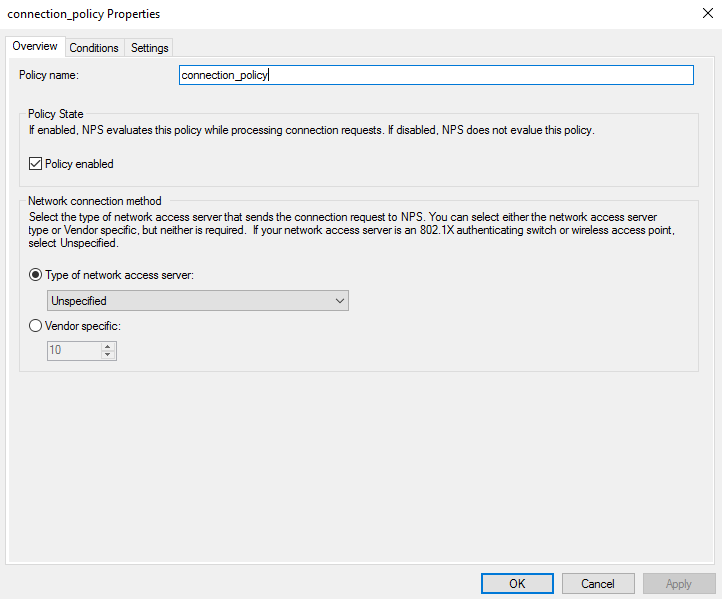
****

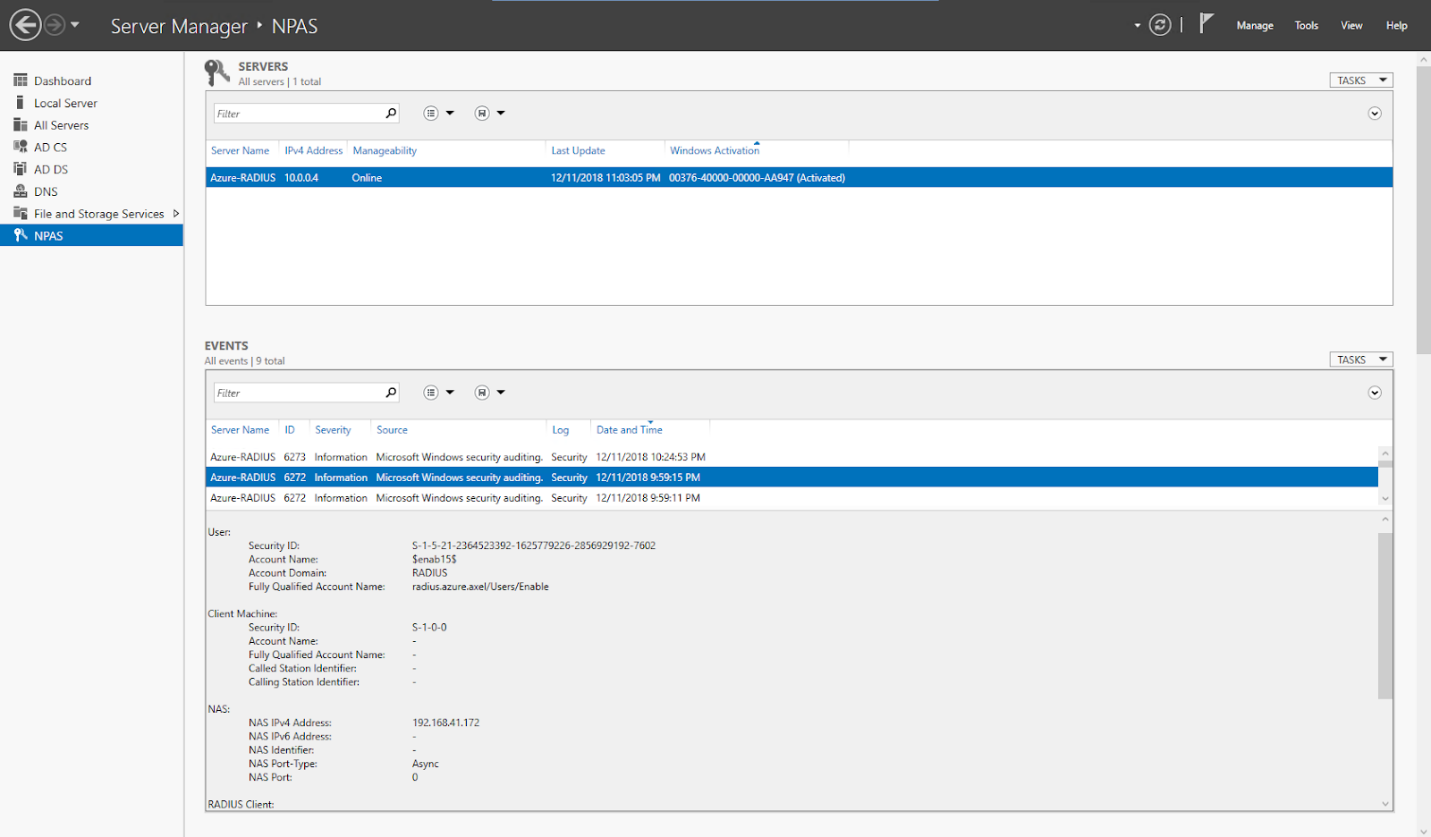
****

****

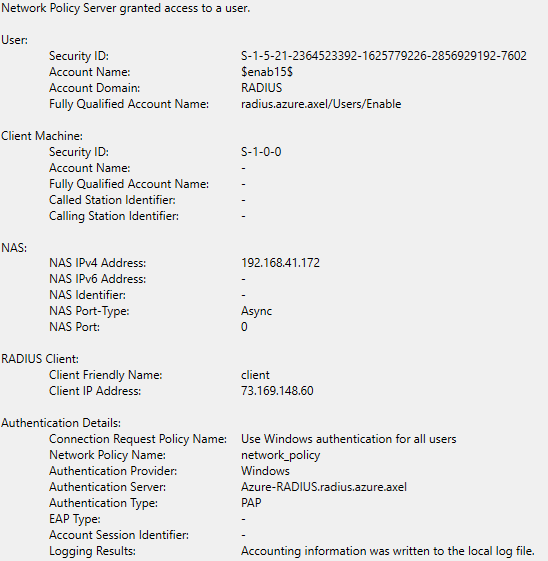
****

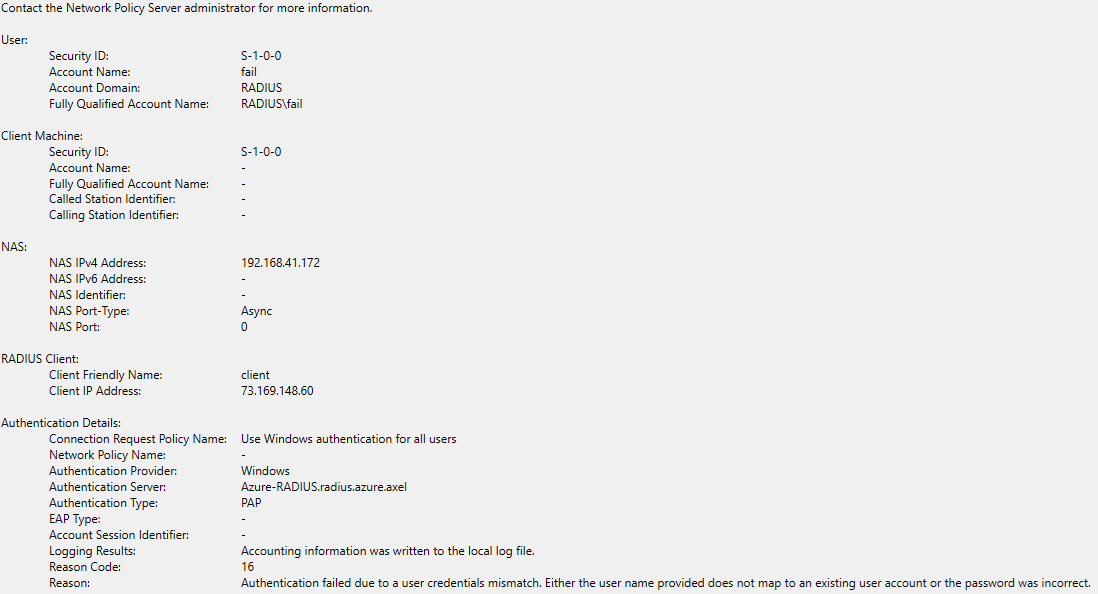
****

****

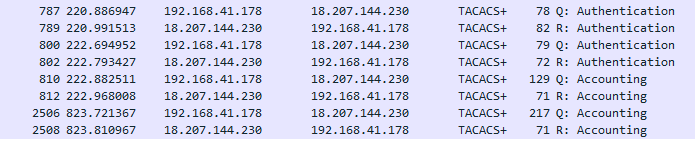
****

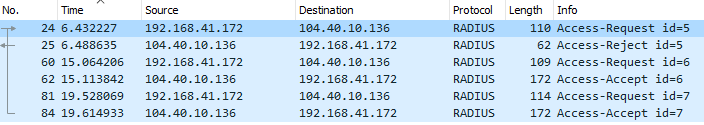
****

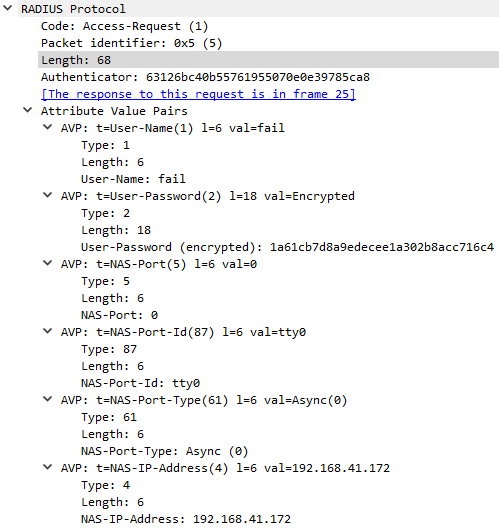
****

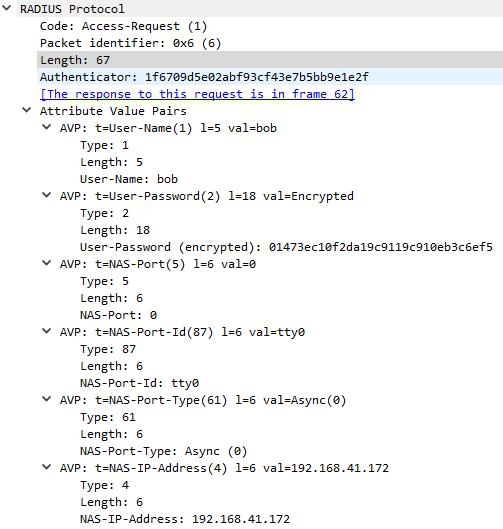
****

Wireshark Captures:

****

****

****

****

# Problems

The lab overall ran smoothly though there were a couple of bumps with the cloud aspect of the lab. We had some difficulties in navigating through AWS as the services that we needed were hidden among the large number of services offered. In addition, we initially did not realize that we had to permit the ports used by TACACS+ (TCP 49) and RADIUS (UDP 1645-1646).

In configuring the RADIUS server, we found we had to set the NAS port types to Async and VPN under the network policy. There were also a couple of other simple mistakes that we made, such as forgetting to allow access under dial-in in user settings and forgetting to add a user to an appropriate group.

Overall there were significantly less issues compared before as we had already practiced configuring TACACS+ and RADIUS in the previous lab, though we did learn quite a few things about working with cloud services.

# Conclusion

The lab was a challenging implementation of TACACS+ and RADIUS authentication through servers online on AWS and Azure. Overall the lab taught us valuable experience with navigating through Azure and AWS and setting up virtual machines and servers on the cloud.