Create an AWS Lab Environment for Demos

Deploy Domain and Active Directory

* Create Cloud Formation stack from AWS Quickstart
* <https://docs.aws.amazon.com/quickstart/latest/active-directory-ds/step2.html>
* Deploy stack under scenario 2
* Test ability to connect to remote desktop gateway
* Modify the *DomainMembers* security group to allow 1433 TCP access on domain

Deploy Windows 2016 AMIs to Domain

* T2.Small or T2.Medium (SQL Server needs memory!)
* Deploy 2 AMI images to private or public subnet 1 from cloud formation stack
* Deploy 1 AMI image to private or public subnet 2 from cloud formation stack
* Add AMI images to the *DomainMembers* security group created by cloud formation stack
* If private connect to images by RDP via the remote desktop gateway already in the domain
* Install SQL Server 2017 Developer Edition
* <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>
* Add AMIs to the domain you specified while building your cloud formation stack
* Change server names, customize to your preferences etc.

Deploy AWS Linux AMI (version 2 min) to Domain

*You can choose another linux distro if you are more comfortable with Ubuntu or Redhat etc.*

* T2.Small or T2.Medium
* Deploy 2 AMI images to public subnet 2 (makes SQL deploy easier from public subnet)
* Add the AMI images to the *DomainMembers* security group created by cloud formation stack
* Install and configure SQL Server using yum
* <https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-overview?view=sql-server-2017>
* Join Linux boxes to your domain
* <https://docs.aws.amazon.com/directoryservice/latest/admin-guide/join_linux_instance.html?icmpid=docs_dirservices_console>

Reduce AWS costs by keeping instances stopped when not actively working with the lab. In my scenario my total AWS costs for the lab, time spent building, creating demos etc, was less than 50$ total. If you are very prudent I could imagine a lab for less than 15$ monthly.

There are guides out on the internet for building labs using both VMWare and Azure as well. I specifically chose AWS simply for the ease of the pre-built AWS Cloud Formation stacks to build out my test domain, with domain controllers, internet access, and multi-subnet environment.

Note from the field:

I recommend turning termination protection on until you are done with the lab, I accidentally deleted one of my domain controllers and wasted a lot of time and effort.

Microsoft SQL Server Always On Availability Groups Checklist

*Node = each individual windows server participating in the cluster  
Replica = each individual sql server instance participating in the group  
Database = each individual database participating in the group*

Configure Always On *(Operating System)*

* Enable windows clustering on each node
* Add participating nodes to cluster
* Validate cluster
* Create cluster
* Multi-subnet?

Configure Availability Groups *(SQL Server)*

* Enable AG on each SQL Server instance through configuration manager
* Create endpoint on each replica
* Grant connect on each endpoint/replica
* Create an Availability Group (primary replica)
* Join each secondary to the new Availability Group (secondary replicas)

Configure Databases *(Database)*

* Join database to AG
* Configure synchronous / asynchronous replication
* Configure manual / automatic failover
* Configure readonly / non-readonly secondary

Configure a Listener *(Availability Group)*

* Select IP for each subnet
* Configure listener
* Test listener
* Test failovers

Configure Advanced Options

* Read only routing
* Offloading backups
* Failover behavior
* Setup monitoring