



ORACLE

# Creating Active Directory Domain Services in Oracle Cloud Infrastructure

---

November 2022, version 3.0  
Copyright © 2022, Oracle and/or its affiliates  
Public

## Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

## Revision History

The following revisions have been made to this document since its initial publication.

DATE	REVISION
November 2022	<ul style="list-style-type: none"><li>• Added design for a multiple-region Active Directory domain</li><li>• Updated Terraform information</li></ul>
November 2021	<ul style="list-style-type: none"><li>• Updated steps and deleted outdated screenshots</li><li>• Added note about Terraform version</li><li>• Updated template</li></ul>
January 2019	Initial publication

# Table of Contents

---

<b>Overview</b>	<b>4</b>
<b>Prerequisites</b>	<b>4</b>
<b>Setting Up the Network</b>	<b>4</b>
Create VCNs	6
Create NAT Gateways	7
Create Private Security Lists	8
Create Security List Rules	8
Create Dynamic Routing Gateways	10
Create Internet Gateways	10
Create Route Tables	11
Create Subnets	12
Create DHCP Rules	15
<b>Creating the Windows Instances</b>	<b>15</b>
<b>Configuring the Active Directory Forest and Domain Controllers</b>	<b>16</b>
Create the Primary Domain Controller	16
Add a Secondary Domain Controller	20
Add Windows Hosts	21
<b>Conclusion</b>	<b>22</b>
<b>Resources</b>	<b>22</b>
<b>Appendix A: ActiveDirectoryInit.ps1</b>	<b>23</b>
<b>Appendix B: ActiveDirectoryInit2.ps1</b>	<b>24</b>
<b>Appendix C: AddComputer.ps1</b>	<b>25</b>
<b>Appendix D: NewComputer.ps1</b>	<b>25</b>

## Overview

Active Directory Domain Services are a proven solution for identity management. Oracle Cloud Infrastructure (OCI) can help you build and extend your current Active Directory forest. This technical paper describes the process of creating an Active Directory environment in an OCI tenancy. Two domain controllers are installed, one active and one read-only, each in a different availability domain for redundancy. A third system is used as a test server to ensure that you can join and log in to the domain established in OCI.

This paper provides the following information:

- Fundamental steps for deploying Active Directory domain controllers
- Best practices for building a simple Active Directory environment and joining domains
- Scripts that you can use to help automate the deployment in an OCI environment

The following topics are out of scope and *not* covered:

- Active Directory design and topologies
- Large forest, tree, and leaf designs
- Group policies or policy management

## Prerequisites

To perform the actions in this paper, you need a nonroot compartment.

You should be familiar with the [fundamentals of the OCI](#). If you haven't used the platform before, try the [getting started tutorial](#).

You should have a basic understanding of [Active Directory concepts](#).

## Setting Up the Network

The following diagrams depict the components of the environment that this paper includes for a single region and for multiple regions. Figure 1 shows the network environment for a single region, Ashburn (ASH).

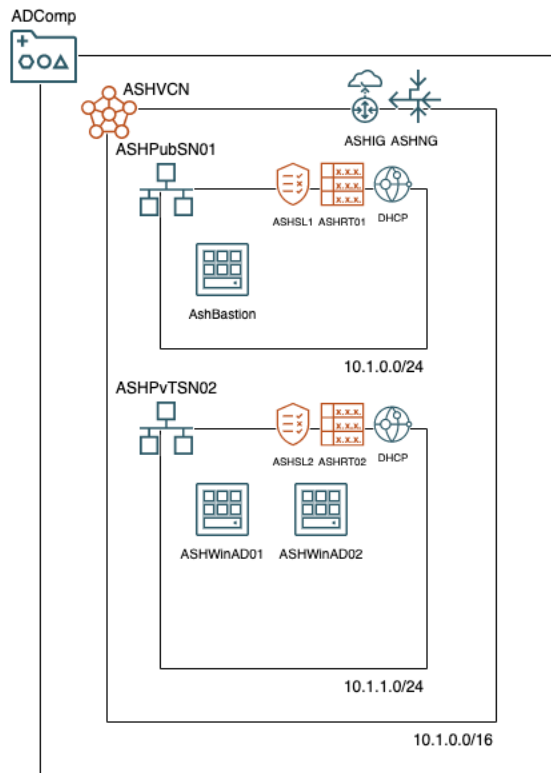


Figure 1: Network Environment for a Single-Region Active Directory

Figure 2 shows the network environment for two regions, Ashburn (ASH) and Phoenix (PHX).

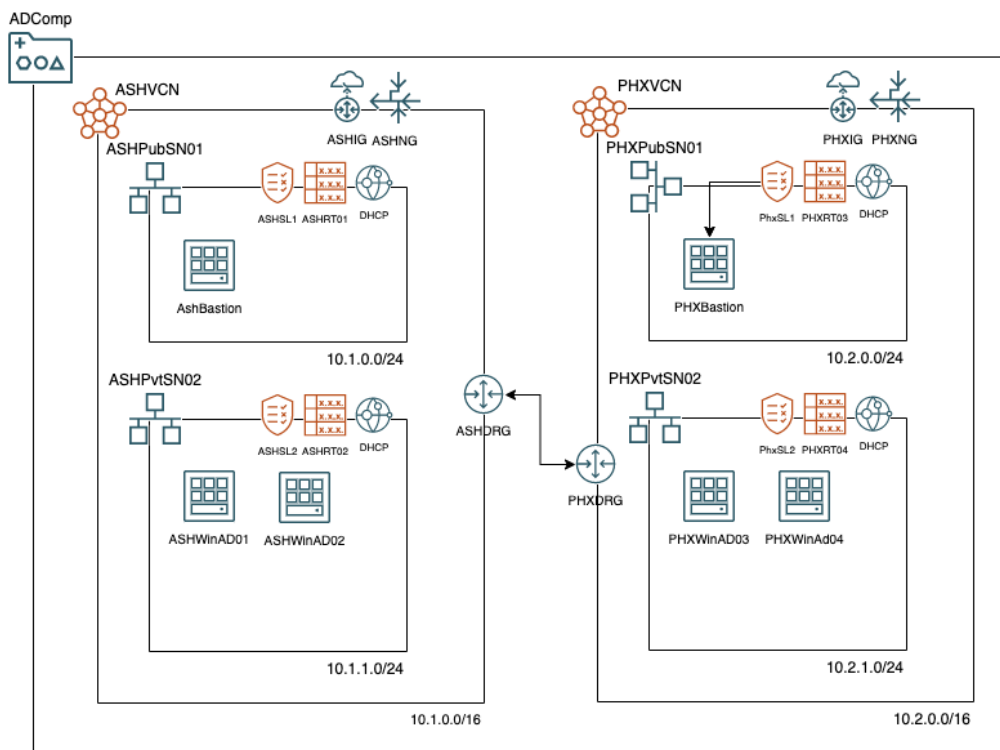


Figure 2: Multiple-Region Network Environment for Active Directory Domain Services

**Best Practice:** The domain controllers shouldn't be accessible externally from the internet. Create one subnet for your domain assets, such as Active Directory domain controllers, and a separate subnet for application servers.

A bastion host is used to access the environment to prevent exposing the remote desktop protocol (RDP) ports of the Active Directory domain controllers to the internet. RDP sessions are tunneled through an SSH connection to a bastion host.

As illustrated in the diagrams, separate subnets are used to host the primary and secondary domain controllers created in the following steps. Because subnets are associated with regions, each domain controller resides in different availability domains, creating an Active Directory domain structure that is resilient to availability domain issues. In the examples that follow, the virtual cloud network (VCN) IP address space 10.1.0.0/16 is used for the Ashburn region and 10.2.0.0/16 is used for the Phoenix region.

---

**Best Practice:** Always be as descriptive as possible when naming OCI components. Descriptive names make it easier when you revisit an environment later.

---

## Create VCNs

Use the Oracle Cloud Console to [create the virtual cloud networks](#) (VCNs) and related resources, including the internet gateway for the bastion host, public route tables, and security lists for the public subnet. Two public subnets are created by default, but they aren't used in this environment. More networking resources are created in the following sections.

---

**Best Practice:** When the architecture has multiple regions, label the environment with meaningful and locational names. Descriptive names help you understand where these items are operating in the environment.

---

Create the following VCNs:

- ASHVCN in the Ashburn region
- PHXVCN in the Phoenix region

The following figure shows the creation of the ASHVCN network in the ADMigration compartment, using 10.1.0.0/16 as the IPv4 CIDR block.

## Create a Virtual Cloud Network [Help](#)

**Name**  
ASHVCN

**Create In Compartment**  
ADMigration  
johnsparkertency (root)/ADMigration

### IPv4 CIDR Blocks

i You can assign up to 5 IPv4 CIDR blocks to a VCN. There must be at least one IPv4 CIDR block assigned to a VCN. [Learn more.](#)

**IPv4 CIDR Blocks**  
10.1.0.0/16 ⌵  
IPv4 Example: 10.0.0.0/16

**DNS Resolution**  
 Use DNS hostnames in this VCN  
Required for instance hostname assignment if you plan to use VCN DNS or a third-party DNS. This choice cannot be changed after the VCN is created. [Learn more.](#)

**DNS Label**  
ASHVCN  
Generated from virtual cloud network name if not specified.

**DNS Domain Name** *Read-Only*  
ASHVCN.oraclevcn.com  
Generated from virtual cloud network name if not specified.

Figure 3: Create a VCN

## Create NAT Gateways

On each VCN, create a NAT gateway to allow the instances that have only private IP addresses to access internet resources.

[Create](#) the following NAT gateways: ASHNG and PHXNG. The following figure shows the creation of the ASHNG gateway in the ASHVCN network.

## Create NAT Gateway [Help](#)

A NAT gateway lets instances that don't have public IP addresses access the internet.

**Name**  
ASHNG

**Create In Compartment**  
ADMigration  
johnsparkertency (root)/ADMigration

**Ephemeral Public IP Address**

The public IP address' lifetime is bound to the lifetime of the NAT Gateway. ✓

**Reserved Public IP Address**

You control the public IP address' lifetime. You can unassign it or re-assign it to another resource in the same region.

Figure 4: Create a NAT Gateway

## Create Private Security Lists

When you create a subnet (which you do in a later section), you must select a security list. Create empty security lists now and then add the rules in the next section.

[Create](#) the following security lists:

- ASHSL1 and ASHSL2 in the ASHVCN network
- PHXSL1 and PHXSL2 in the PHXVCN network

The following figure shows the creation of the ASHSL1 security list in the ASHVCN network.

The screenshot shows the 'Create Security List' form. At the top, there is a 'Name' field containing 'ASHSL1'. Below it is a 'Create In Compartment' dropdown menu showing 'ADMigration'. There are two sections for rules: 'Allow Rules for Ingress' and 'Allow Rules for Egress', each with a '+ Another Ingress Rule' or '+ Another Egress Rule' button. At the bottom, there are fields for 'Optional tags to organize and track resources' with 'Tag Namespace', 'Tag Key', and 'Tag Value' dropdowns and input fields, and a '+ Another Tag' button. The form has 'Create Security List' and 'Cancel' buttons at the bottom.

Figure 5: Create a Security List

## Create Security List Rules

Active Directory uses several protocols to communicate, including RPC, NetBIOS, SMB, LDAP, Kerberos, WINS, and DNS. All the protocols are listed in the following table, although your configuration might use only some of them. If a protocol, such as WINS, isn't used in your environment, you can remove it from the list.

As a best practice, place all the domain controllers in a subnet that either has no external IP addresses or has no access from the internet. As a result, you might want to enable all ports to communicate between your subnets and the Active Directory subnets. However, this action still opens potential paths of attack from those subnets. So, it's a best practice to open only the following ports between the subnets.

NAME	PROTOCOL	PORT
RDP	TCP	3389
DNS	TCP, UDP	53



NAME	PROTOCOL	PORT
LDAP	TCP, UDP	389
LDAP over SSL	TCP	636
Global catalog LDAP	TCP	3268
Global catalog LDAP over SSL	TCP	3269
Kerberos	TCP, UDP	88
RPC endpoint mapper	TCP, UDP	135
NetBIOS name service	TCP, UDP	137
NetBIOS datagram service	UDP	138
NetBIOS session service	TCP	139
SMB over IP (Microsoft-DS)	TCP, UDP	445
WINS resolution	TCP, UDP	1512
WINS replication	TCP, UDP	42

[Create](#) ingress rules on all the security lists to allow the required port communication into the new Active Directory subnets (these rules must exist to allow traffic between the domain controller subnets). The following figures show the creation of two ingress rules in the ASHSL1 security list, both for Kerberos, one using TCP and the other using UDP.

**Ingress Rule 9**

Allows TCP traffic 88

Stateless ⓘ

Source Type: CIDR

Source CIDR: 10.0.1.0/24  
Specified IP addresses: 10.0.1.0-10.0.1.255 (256 IP addresses)

IP Protocol: TCP

Source Port Range: All (Optional ⓘ)  
Examples: 80, 20-22

Destination Port Range: 88 (Optional ⓘ)  
Examples: 80, 20-22

Description: Kerberos (Optional ⓘ)  
Maximum 255 characters

Figure 6: Add a Security List Rule for Kerberos on TCP

**Ingress Rule 10**

Allows UDP traffic 88

Stateless ⓘ

Source Type: CIDR | Source CIDR: 10.0.1.0/24 | IP Protocol: UDP

Specified IP addresses: 10.0.1.0-10.0.1.255 (256 IP addresses)

Source Port Range: All | Destination Port Range: 88

Examples: 80, 20-22

Description: Kerberos

Maximum 255 characters

Figure 7: Add Security List Rule for Kerberos on UDP

## Create Dynamic Routing Gateways

[Create](#) dynamic routing gateways (DRGs) to connect each regional VCN, which allows traffic between the regions. Ensuring that the domain controllers can talk with each other also ensures that they can replicate data between controllers, which ensures access to the active directory domain.

The following figure shows the creation of the dynamic routing gateway ASHDRG in the ASHVCN network.

**Create DRG Attachment**

Name: ASHDRG

DRG Location:  Current tenancy  Another tenancy

Choose a DRG in **ADMigration** (Change Compartment)

No data available

Hide Advanced Options

VCN Route Table Association: Tags

Use this advanced feature only if you're setting up [transit routing](#).

Associate Vcn Route Table:  None  Select Existing

VCN Route Table Association in **ADMigration** (Change Compartment): ASHRT02

Figure 8: Create a Dynamic Routing Gateway

## Create Internet Gateways

Internet gateways route traffic to the internet for the public subnets. [Create](#) the following internet gateways:

- ASHIG in the ASHVCN network
- PHXIG in the PHXVCN network

The following figure shows the creation of the ASHIG internet gateway.

**Create Internet Gateway** [Help](#)

Name  
ASHIG

Create In Compartment  
ADMigration  
johnsparkertency (root)/ADMigration

[Show Advanced Options](#)

Figure 9: Create Internet Gateways

## Create Route Tables

Create route tables to use for the subnets that you create in a later section. Private subnets can automatically route to other private subnets in the VCN. The NAT gateway that you created is used by this route table for all internet destinations, which allows instances that have only private IP addresses to access internet resources.

[Create](#) the following route tables:

- ASHRT01 for the ASHVCN public subnet, and ASHRT02 for the ASHVCN private subnet
- PHXRT03 for the PHXVCN public subnet, and PHXRT04 for the PHX VCN private subnet

Create the route tables with a 0.0.0.0/0 route to the NAT gateway (for the private subnets) and the internet gateway (for the public subnets). Figure 10 shows the creation of the ASHRT02 route table, and Figures 11 and 12 show the route rules created for the ASHRT02 and ASHRT01 route tables.

**Create Route Table** [Help](#)

Name  
ASHRT02

Create In Compartment  
ADMigration  
johnsparkertency (root)/ADMigration

**Route Rules (Optional)**


**Important:**  
For a route rule that targets a Private IP, you must first enable "Skip Source/Destination Check" on the VNIC that the Private IP is assigned to.

[+ Another Route Rule](#)

[Show Tagging Options](#)

Figure 10: Create Route Tables

Networking » Virtual Cloud Networks » ASHVCN » Route Table Details



AVAILABLE

## ASHRT02

Move resource Add Tags Terminate

Route Table Information Tags

OCID: ...df4oxa [Show](#) [Copy](#) Compartment: ADMigration

Created: Fri, May 6, 2022, 18:37:48 UTC

Resources

[Route Rules \(1\)](#)

### Route Rules

Traffic within the VCN is handled by the VCN's local routing by default. Intra-VCN routing allows you more control over routing between subnets. [Learn more](#)


Add Route Rules Edit Remove

<input type="checkbox"/>	Destination	Target Type	Target	Description
<input type="checkbox"/>	0.0.0.0/0	NAT Gateway	<a href="#">ASHNG</a>	

0 Selected

Figure 11: Route Rules for the NAT Gateway on Private Subnets

Networking » Virtual Cloud Networks » ASHVCN » Route Table Details



AVAILABLE

## ASHRT01

Move resource Add Tags Terminate

Route Table Information Tags

OCID: ...m3omuq [Show](#) [Copy](#) Compartment: ADMigration

Created: Fri, May 6, 2022, 18:36:27 UTC

Resources

[Route Rules \(1\)](#)

### Route Rules

Traffic within the VCN is handled by the VCN's local routing by default. Intra-VCN routing allows you more control over routing between subnets. [Learn more](#)

Add Route Rules Edit Remove

<input type="checkbox"/>	Destination	Target Type	Target	Description
<input type="checkbox"/>	0.0.0.0/0	Internet Gateway	<a href="#">ASHIG</a>	

0 Selected

Figure 12: Route Table Rules for the Internet Gateway on Public Subnets

## Create Subnets

In this architecture, you create a private subnet for the Active Directory environment and a public subnet for the bastion or RDP service. If you plan to combine the OCI network with on-premises or external networks, then you can join access with a VPN, which isn't shown here. Subnets are available across a region.

[Create](#) the following subnets, using the route tables and security lists that you already created.

SUBNET NAME	CIDR BLOCK	ROUTE TABLE	SECURITY LIST
ASHPubSN01	10.1.0.0/24	ASHRT01	ASHSL1
ASHPvtSN02	10.1.1.0/24	ASHRT02	ASHSL2
PHXPubSN01	10.2.0.0/24	PHXRT03	PhxSL1
PHXPvtSN02	10.2.1.0/24	PHXRT04	PhxSL2

The following figure shows the creation of the regional ASHPubSN01 public subnet in the ADMigration compartment, with the values listed in the preceding table.

**Create Subnet**

Name  
ASHPubSN01

Create In Compartment  
ADMigration  
johnsparkertency (root)/ADMigration

Subnet Type  
**Regional (Recommended)**  
Instances in the subnet can be created in any availability domain in the region. Useful for high availability. ✓  
Availability Domain-specific  
Instances in the subnet can only be created in one availability domain in the region.

IPv4 CIDR Block  
IPv4 CIDR Block  
10.1.0.0/24  
Specified IP addresses: 10.1.0.0-10.1.0.255 (256 IP addresses)

IPv6 Prefixes  
Maximum amount of one IPv6 prefix per subnet. [Learn more.](#)

Route Table Compartment in **ADMigration** ([Change Compartment](#))  
ASHRT01

Subnet Access  
**Private Subnet**  
Prohibit public IP addresses for Instances in this Subnet  
**Public Subnet**  
Allow public IP addresses for Instances in this Subnet ✓

DNS Resolution  
 Use DNS hostnames in this SUBNET ⓘ  
Allows assignment of DNS hostname when launching an Instance

DNS Label  
ASHPubSN01  
Only letters and numbers, starting with a letter. 15 characters max.

DNS Domain Name *Read-Only*  
<dns-label>.ashvcn.oraclevcn.com

Dhcp Options Compartment in **ADMigration** ([Change Compartment](#))  
Default DHCP Options for ASHVCN

Security Lists  
You can associate up to 5 network security lists with the subnet.  
Security List Compartment in **ADMigration** ([Change Compartment](#))  
ASHSL1  
[+ Another Security List](#)

[Show Tagging Options](#)

[Create Subnet](#) [Cancel](#)

Figure 13: Create a Public Subnet

The following figure shows the creation of the regional ASHPvtSN02 private subnet in the ADMigration compartment, with the values listed in the preceding table.

### Create Subnet

Name  
ASHPvtSN02

Create In Compartment  
ADMigration  
johnsparkertency (root)/ADMigration

Subnet Type

**Regional (Recommended)**  
Instances in the subnet can be created in any availability domain in the region. Useful for high availability. ✓

Availability Domain-specific  
Instances in the subnet can only be created in one availability domain in the region.

IPv4 CIDR Block  
IPv4 CIDR Block  
10.1.1.0/24  
Specified IP addresses: 10.1.1.0-10.1.1.255 (256 IP addresses)

IPv6 Prefixes  
**i** Maximum amount of one IPv6 prefix per subnet. [Learn more.](#)

Route Table Compartment in **ADMigration** [\(Change Compartment\)](#)  
ASHRT02

Subnet Access

**Private Subnet**  
Prohibit public IP addresses for Instances in this Subnet ✓

Public Subnet  
Allow public IP addresses for Instances in this Subnet

DNS Resolution  
 Use DNS hostnames in this SUBNET ⓘ  
Allows assignment of DNS hostname when launching an Instance

DNS Label  
ASHPvtSN02  
Only letters and numbers, starting with a letter. 15 characters max.

DNS Domain Name *Read-Only*  
<dns-label>.ashvcn.oraclevcn.com

Dhcp Options Compartment in **ADMigration** [\(Change Compartment\)](#)  
Default DHCP Options for ASHVCN

Security Lists  
You can associate up to 5 network security lists with the subnet.  
Security List Compartment in **ADMigration** [\(Change Compartment\)](#)  
ASHSL2  
[+ Another Security List](#)

[Show Tagging Options](#)

[Create Subnet](#) [Cancel](#)

Figure 14: Create a Private Subnet

## Create DHCP Rules

[Create](#) specific DHCP rules to make instances use the domain controllers as the primary DNS. These rules make the Active Directory the main source of DNS traffic. The Active Directory DNS points to OCI for outbound resolution. You configure these rules after the Active Directory controllers are established.

## Creating the Windows Instances

The example in this paper uses Windows Server 2019 instances. In each VCN, two instances are used for the Active Directory domain controllers, and a third is joined to the domain as a bastion host. Use the following properties when you create the instances in the following section. (The shape used in this paper is a recommendation; scale it up or down as needed).

NAME	IMAGE	SHAPE	CORES	AVAILABILITY DOMAIN	SUBNET
ASHWinAD01	Windows Server 2019 Standard VM	VM.Standard.E4.Flex	1	ASH-AD-1	ASHPvtSN02
ASHWinAD02	Windows Server 2019 Standard VM	VM.Standard.E4.Flex	1	ASH-AD-2	ASHPvtSN02
ASHBastion	Windows Server 2019 Standard VM	VM.Standard.E4.Flex	1	ASH-AD-3	ASHPubSN01
PHXWinAD03	Windows Server 2019 Standard VM	VM.Standard.E4.Flex	1	PHX-AD-1	PHXPvtSN02
PHXWinAD04	Windows Server 2019 Standard VM	VM.Standard.E4.Flex	1	PHX-AD-2	PHXPvtSN02
PHXBastion	Windows Server 2019 Standard VM	VM.Standard.E4.Flex	1	PHX-AD-3	PHXPubSN01

For each instance, note the RFC1918 IP addresses.

INSTANCE	RFC1918 IP
ASHWinAD01	10.1.1.2
ASHWinAD02	10.1.1.3
ASHBastion	10.1.0.20
PHXWinAD03	10.2.1.2
PHXWinAD04	10.2.1.3
PHXBastion	10.2.0.20

## Configuring the Active Directory Forest and Domain Controllers

You can create your initial domain controller in several different ways. This paper uses Microsoft PowerShell integrated with [cloudbase-init](#) to reduce the amount of manual interaction with the Active Directory setup. The scripts provided in the appendices install the necessary Windows Server features, such as the .NET Framework, Active Directory Domain Services, and the DNS server components. We use the following PowerShell scripts to create this environment.

LOCATION	SCRIPT NAME	DESCRIPTION
Appendix A: ActiveDirectoryInit.ps1	ActiveDirectoryInit.ps1	Create the forest and promote the server to an Active Directory domain controller.
Appendix B: ActiveDirectoryInit2.ps1	ActiveDirectoryInit2.ps1	Build the second host and promote it to be the replica domain controller.
Appendix C: AddComputer.ps1	AddComputer.ps1	Prepare the domain for a new computer join.
Appendix D: NewComputer.ps1	NewComputer.ps1	Join a Windows Server to the domain at deployment time.

This paper uses the Oracle Cloud Console to demonstrate how to create the compute instances. You need the following information:

- Your domain administrator password. As a best practice, ensure that you change your domain administrator password immediately after you create the domain controllers.
- The name of the domain that you create.
- A one-time password that you use when joining new computers to the domain.

### Create the Primary Domain Controller

This procedure shows creating the ASHWinAD01 instance.

1. In the Console navigation menu, click **Compute** and then click **Instances**.
2. Click **Create instance**.
3. Provide a name for the instance and select the compartment to create it in.

### Create compute instance

Create an instance to deploy and run applications, or save as a reusable Terraform stack for creating an instance with Resource Manager.

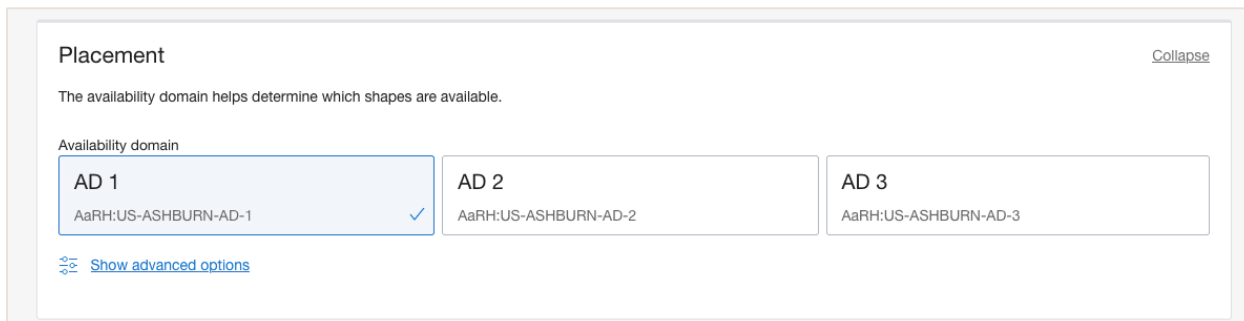
Name

Create in compartment

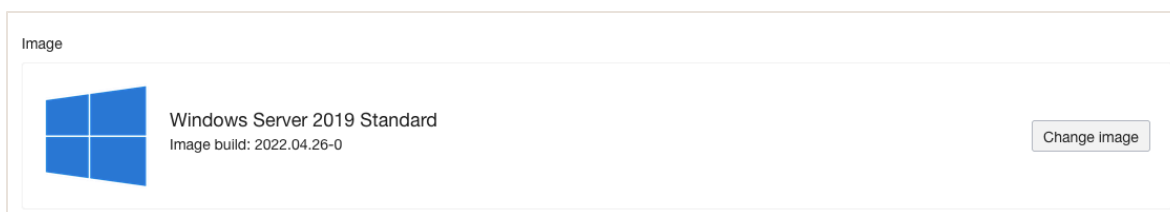
johnsparkertency (root)/ADMigration



4. Select the availability domain.



5. Click **Show advanced options**, and select the fault domain.
6. Choose the image operating system (Windows Server 2019 Standard) and version.



7. For the shape, choose the instance type (virtual machine) and the instance shape (VM.Standard.E4.Flex).



---

**Note:** You can choose a larger boot volume size, or you can encrypt the boot volume through the OCI Vault service. This page doesn't address this function.

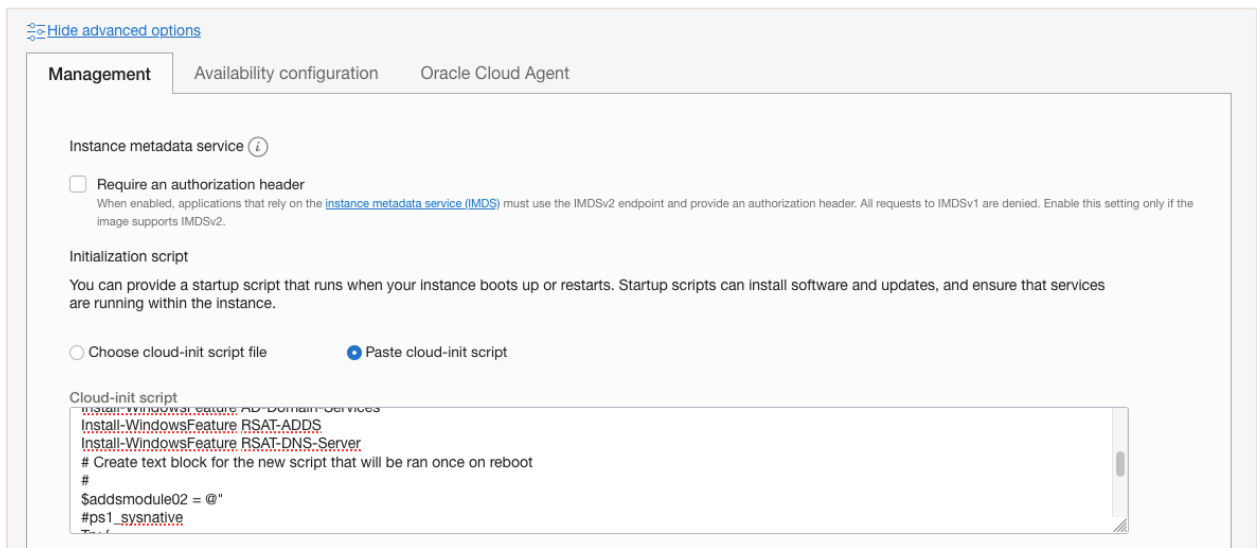
---

8. Configure the networking connection. For example:
  - o Compartment: **ADMigration**
  - o VCN: **ASHVCN**
  - o Subnet: **ASHPvtSN02**

**Best Practice:** Ensure that the new domain controllers are in a private subnet.

9. At the bottom of the page, click **Show advanced options**.
10. On the **Management** tab, select **Paste cloud-init script**.

- Copy the `ActiveDirectoryInit.ps1` script from “Appendix A: ActiveDirectoryInit.ps1” and paste it in the **Cloud-init script** text box.

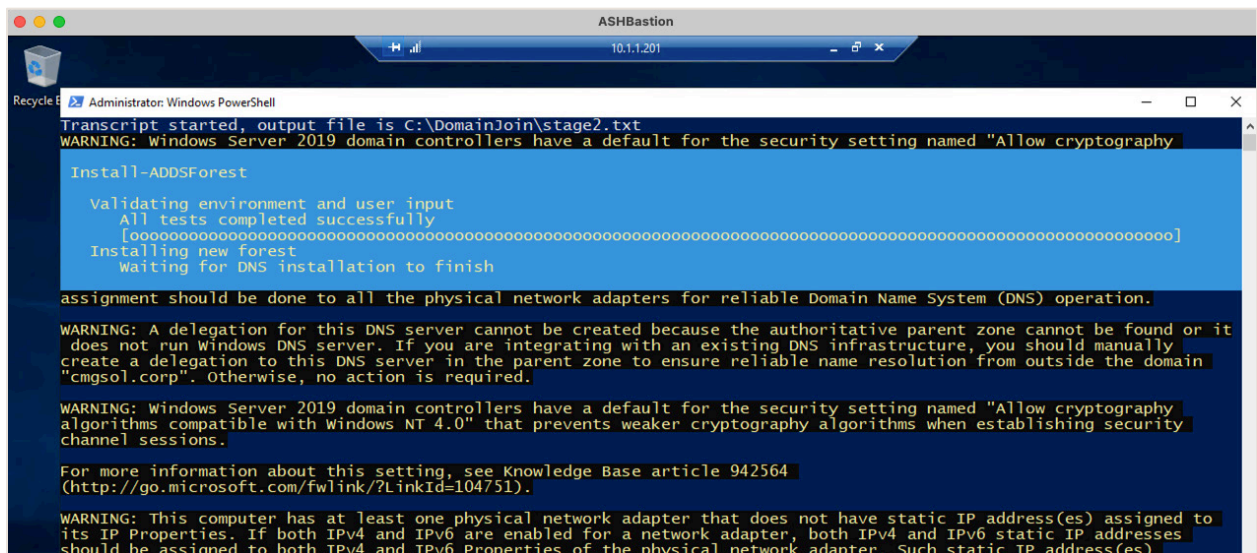


- Click **Create**.

The script takes some time to complete the installation of the Windows features and the Active Directory tools.

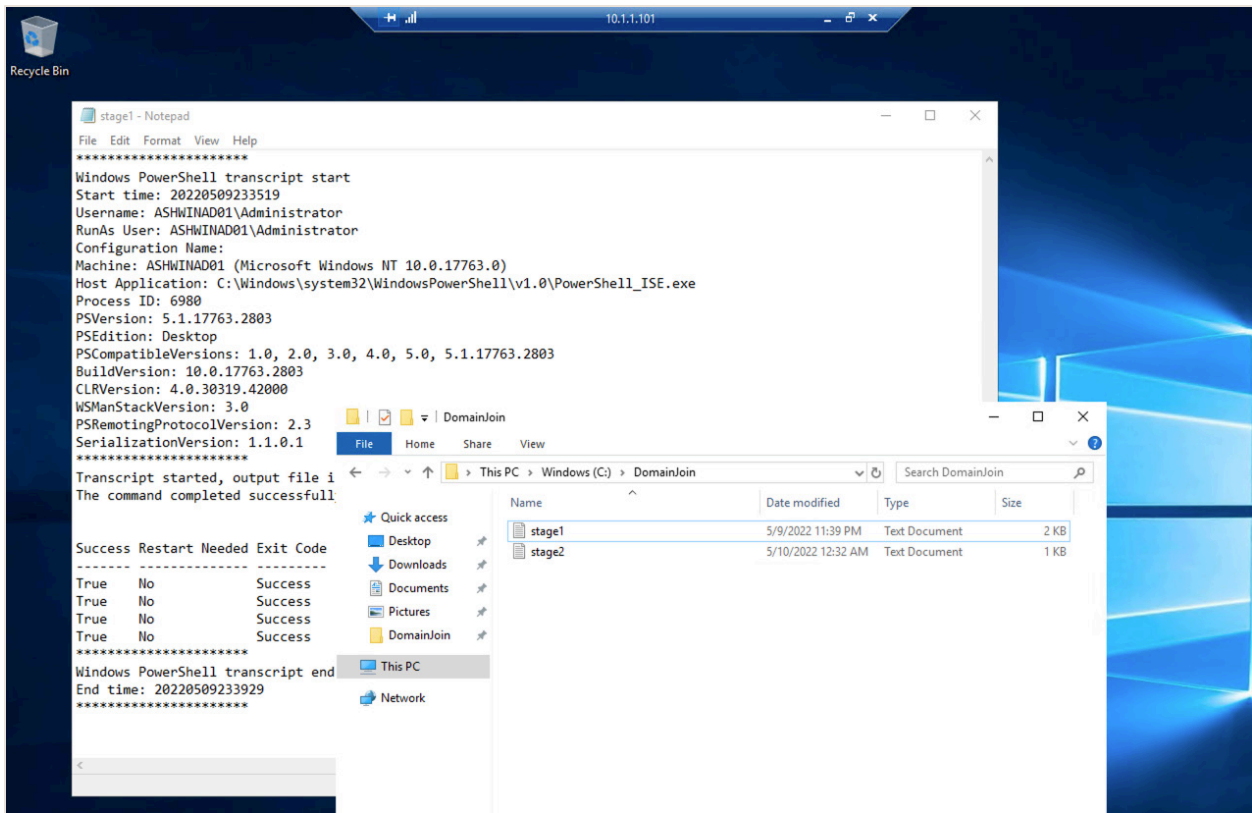
You can log in and monitor the progress by viewing the `stage1.txt` file at `c:\DomainJoin\stage1.txt`. The log should show `Success = True` for the .NET Framework, Active Directory Domain Services, Active Directory Administrative Center, and DNS server tools.

- After the first restart, log in to the host with the domain administrator account to run the last script with the `RunOnce` script. The first login to the host with the domain administrator account starts the `RunOnce` script and provides a reference when the entire process is done.

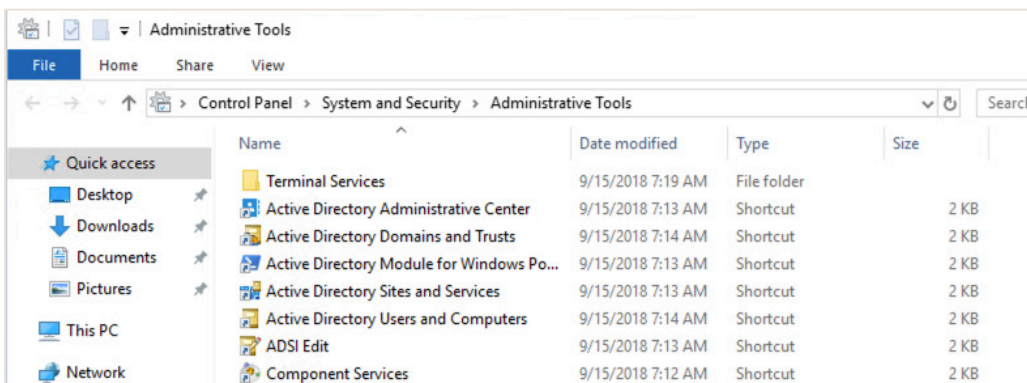


After the `RunOnce` script runs, the instance restarts automatically as part of the process.

- Log back in as the domain admin and check the logs to ensure that no errors occurred. The log files are stage1.txt and stage2.txt in the C:\DomainJoin directory. For success, stage2.txt can have warnings but no errors.



- Verify that the domain has been successfully created by opening the **Start** menu, selecting **Windows Administrative Tools**, and then selecting **Active Directory Users and Computers**.



Now, you have the first domain controller in the new Active Directory forest. The new forest is ready for configuration that isn't covered in this paper, such as group policies, more domain trusts, and DNS configurations.

## Add a Secondary Domain Controller

1. Repeat steps 1–9 in the previous section to create a backup domain controller. Make the appropriate changes in the name of the instance and in setting the appropriate [availability domain and fault domain](#) to ensure that you have proper redundancy for the domain. The next steps use the script from “Appendix B: ActiveDirectoryInit2.ps1.”  
ActiveDirectoryInit2.ps1.”

**Best Practice:** To ensure the best availability, deploy across multiple availability domains or fault domains within one availability domain.

2. On the **Management** tab, select **Paste cloud-init script**.
3. Copy the ActiveDirectoryInit2.ps1 script from “Appendix B: ActiveDirectoryInit2.ps1” and paste it in the **Cloud-init script** text box.
4. In the script, adjust the \$DnsServer variable to the private IP address of the primary domain controller that you previously created.

```
$DnsServer = 'private IP address for current domain controller'
```

Hide advanced options

Management Availability configuration Oracle Cloud Agent

Instance metadata service ⓘ

Require an authorization header

When enabled, applications that rely on the [instance metadata service \(IMDS\)](#) must use the IMDSv2 endpoint and provide an authorization header. All requests to IMDSv1 are denied. Enable this setting only if the image supports IMDSv2.

Initialization script

You can provide a startup script that runs when your instance boots up or restarts. Startup scripts can install software and updates, and ensure that services are running within the instance.

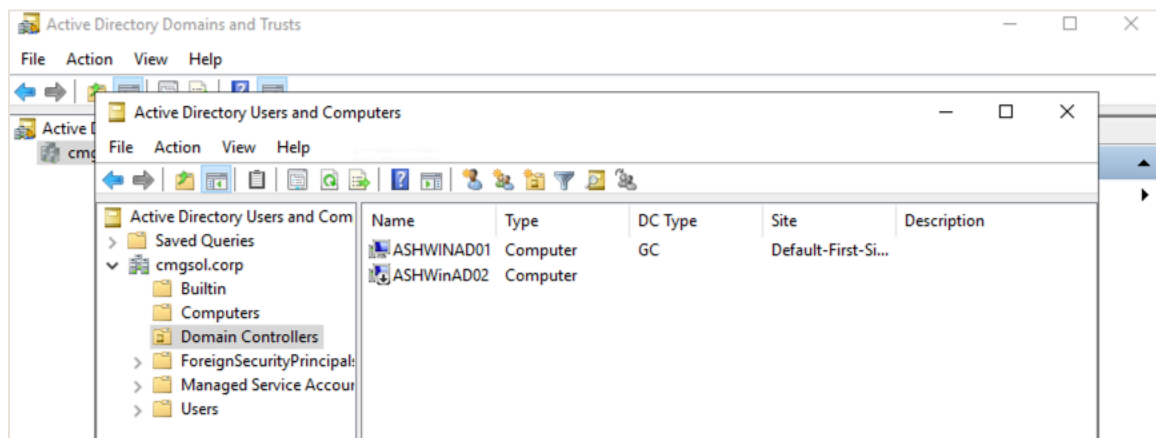
Choose cloud-init script file  Paste cloud-init script

Cloud-init script

```
$DomainUser='cmgsofadministrator'
$EncryptedPass = ConvertTo-SecureString $Password -AsPlainText -Force
$Credential = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList $DomainUser, $EncryptedPass
# Enter the address for the domain controller.
$DnsServer = '10.1.1.101'
#Set the Administrator Password and activate the Domain Admin Account
net user Administrator $Password /logonpasswordchg:no /active:yes
```

5. Click **Create**.

You can monitor the progress by watching the **Domain Controllers** section in the current domain controller under **Active Directory Users and Computers**. It takes approximately 20 minutes to install all the necessary Windows Server features and add the server to the domain.



6. After the final restart of the host, check the installation by logging in with the domain administrator account. Check the `C:\DomainJoin\stage3.txt` log file. Also verify that the Active Directory tools are loaded on the host.

The `stage3.txt` file should show `Success = True` for the .NET Framework, Active Directory Domain Services, Active Directory Administrative Center, and DNS server tools. You should see only warnings and no errors, success for the DC promo, and that a restart is required. The script restarts the host after five minutes.

7. To verify communication with the domain, run the `Get-ADForest` command from the PowerShell command prompt.

The output shows the correct domains and name for the Active Directory domain and forest.

8. After the domain controllers are installed, change the domain administrator password by using the `Set-ADAccountPassword` command. Ensure that you use a strong password that meets the password standards of your organization.

---

**Caution:** Skipping this step can create a security threat to your Active Directory domain.

---

You now have a primary Active Directory domain controller and a secondary domain controller to facilitate a complete Active Directory forest in your OCI tenancy. Add any of the group policies and users that you require in your environment.

## Add Windows Hosts

Now you can add hosts to the domain. You can join new computers to the Active Directory domain in many ways. This paper uses a Microsoft PowerShell example for using a predefined computer credential to add a host to the domain. The Microsoft website has more examples that you can use to add hosts to a domain.

---

**Best Practice:** Use the [Microsoft PowerShell example](#) of using a predefined computer credential to add hosts to your domain.

---

1. Log in to the primary domain controller with a domain administrator account.
2. Open a PowerShell window and run the `AddComputer.ps1` script from “Appendix C: AddComputer.ps1.”  
The script runs the `New-ADComputer` command to add the new computer record in the domain controller.
3. Verify that the computer was added by checking the **Computers** section in **Active Directory Users and Computers**.

After you add the computer, you can create the instance in your OCI tenancy.

4. Sign in to the Oracle Cloud Console and create the Windows Server 2019 instance by following steps 1–9 in the “Create the Primary Domain Controller” section of this paper. Enter an appropriate name, place the instance in the correct availability and fault domains, and pick a subnet and shape that are correct for your needs.
5. On the **Management** tab, select **Paste cloud-init script**.
6. Copy the `NewComputer.ps1` script from “Appendix D: NewComputer.ps1” and paste it in the **Cloud-init script** text box.
7. In the script, update the `$DnsServer` variable with the correct IP address for the domain controllers.

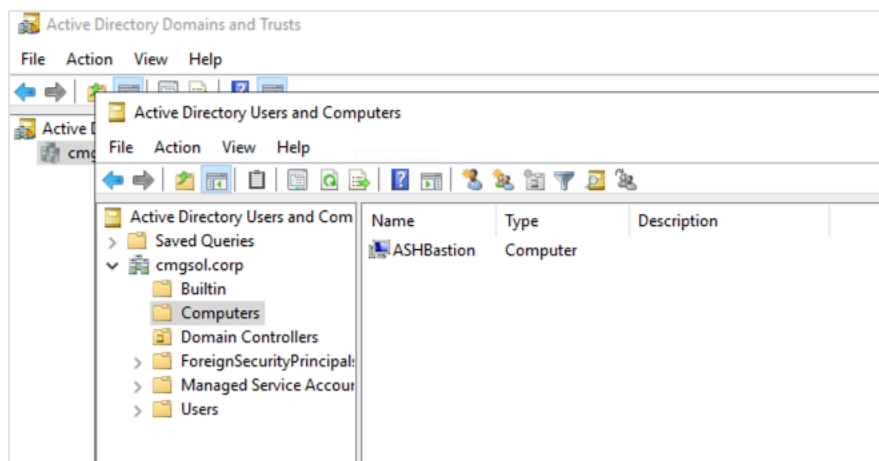


8. Click **Create**.

After the computer has joined the domain, it automatically restarts. The script contains a five-minute sleep to ensure that domain replication has occurred.

9. Log in to the new host and check the C:\DomainJoin\Stage4.txt log file for errors.

You can also check the computer properties of the host in the **Active Directory Users and Computers** administrative application.



Now you have a fully functioning Active Directory domain to which you can add more computers and expand your domain to fit the needs of your organization.

## Conclusion

This paper walks through the core steps of building an Active Directory domain, using redundant domain controllers in separate OCI availability or fault domains and logical subnets to ensure that you're building fault tolerance into your infrastructure. You can build more application servers to add to the domain. These servers are the building blocks of your Active Directory domain. It's up to you to build your group policies and ensure that your domain meets the standards of your organization.

Oracle Cloud Infrastructure enables you to deploy the building blocks of your Active Directory domain and support any expansion to the Active Directory forests that your organization requires to meet the demanding needs of today's computing environments.

## Resources

- [Oracle Cloud Infrastructure documentation](#)
- [OCI regions and availability domains](#)
- [Creating an OCI VCN](#)
- [Bastion Hosts: Protected Access for VCNs](#)
- [Microsoft Active Directory Services overview](#)
- [PowerShell: Add-Computer](#)
- [PowerShell: Active Directory Commands](#)
- [PowerShell: RunOnce Registration Key](#)
- [PowerShell: General documentation](#)

## Appendix A: ActiveDirectoryInit.ps1

```
#ps1_sysnative
#####
# Title: ActiveDirectoryInit.ps1
# Version & Date: v1 31 Oct 2018
# Updated: v2 31 Mar 2022
# Creator: john.s.parker@oracle.com
# Warning: This script is a representation of how to use PowerShell to create an Active Directory Domain controller
#          and build the first DC in a new Active Directory Forest. This script creates and uses the domain administrator account
#          there are potential for mistakes and destructive actions. USE AT YOUR OWN RISK!!
# This is the first script in the Active Directory Series that will establish the first
# Active Directory Domain Controller. This script will unlock the local administrator account
# this account will become the Domain Administrator.
#
# This script will install the required Windows features that are required for Active
# Directory. This script will install the prerequisites for Active Directory, then create a
# one-time executed script on the login after the reboot. This script will reboot the host
# a total of 2 times to add the windows features, create the forest, and promote the domain controller.
#
# Variables for this script
# $password - this is the password necessary to unlock the administrator account
#            - and is used in both runs of the AD build.
# $FullDomainName - the full name for the AD Domain example: CESA.corp
# $ShortDomainName - the short name for the AD Domain example: CESA
# $encrypted - you must encrypt the password so that you can use it as you set up your domain controller
# $addsmodule02 - this is the text block that will be used to create the RunOnceScript that will finish the installation
#                - of the domain controller.
# $RunOnceKey - this is the key that will create the command to complete the installation of the domain controller.
Try {
#
# Start the logging in the C:\DoimainJoin directory
#
Start-Transcript -Path "C:\DomainJoin\stage1.txt"
# Global Variables
$password="Password!!"
# Set the Administrator Password and activate the Domain Admin Account
#
net user Administrator $password /logonpasswordchg:no /active:yes
# Install the Windows features necessary for Active Directory
# Features
# - .NET Core
# - Active Directory Domain Services
# - Remote Active Directory Services
# - DNS Services
#
Install-WindowsFeature NET-Framework-Core
Install-WindowsFeature AD-Domain-Services
Install-WindowsFeature RSAT-ADDS
Install-WindowsFeature RSAT-DNS-Server
# Create text block for the new script that will be ran once on reboot
#
$addsmodule02 = @"
#ps1_sysnative
Try {
Start-Transcript -Path C:\DomainJoin\stage2.txt
$password = "Password!!"
`$FullDomainName = "cmgsol.corp"
`$ShortDomainName = "CMGSOL"
`$encrypted = ConvertTo-SecureString `password -AsPlainText -Force
Import-Module ADDSDeployment
Install-ADDSForest `
-CreateDnsDelegation:`$false `
-DatabasePath "C:\Windows\NTDS" `
-DomainMode "WinThreshold" `
-DomainName `$FullDomainName `
-DomainNetbiosName `$ShortDomainName `
-ForestMode "WinThreshold" `
-InstallDns:`$true `
-LogPath "C:\Windows\NTDS" `
-NoRebootOnCompletion:`$false `
-SysvolPath "C:\Windows\SYSVOL" `
-SafeModeAdministratorPassword `$encrypted `
-Force:`$true
} Catch {
Write-Host $_
} Finally {
```

```

Stop-Transcript
}
"@
Add-Content -Path "C:\DomainJoin\ADDModule2.ps1" -Value $addsmodule02
# Adding the run once job
#
$RunOnceKey = "HKLM:\Software\Microsoft\Windows\CurrentVersion\RunOnce"
set-itemproperty $RunOnceKey "NextRun" ('C:\Windows\System32\WindowsPowerShell\v1.0\Powershell.exe -executionPolicy Unrestricted -File ' +
"C:\DomainJoin\ADDModule2.ps1")
# End the logging
#
} Catch {
Write-Host $_
} Finally {
Stop-Transcript
}
# Last step is to reboot the local host
#
Restart-Computer -ComputerName "localhost" -Force

```

## Appendix B: ActiveDirectoryInit2.ps1

```

#ps1_sysnative
#####
# Title: ActiveDirectoryInit2.ps1
# Version & Date: v1 31 Oct 2018
# Updated: v2 31 Mar 2022
# Creator: john.s.parker@oracle.com
# Warning: This script is a representation of how to use PowerShell to create an Active Directory Domain controller
#          and build the first DC in a new Active Directory Forest. This script creates and uses the domain administrator account
#          there are potential for mistakes and destructive actions. USE AT YOUR OWN RISK!!
# This is the second script in the Active Directory Series that will establish the second
# Active Directory Domain Controller. This script will unlock the local administrator account.
#
# This script will install the required Windows features that are required for Active
# Directory. This script will install the prerequisites for Active Directory. This script will reboot the host after it has added
# the
# Windows features installed the Active Directory Services and promoted the domain controller.
#
# Variables for this script
# $password - this is the password necessary to unlock the administrator account
#           - and is used in both runs of the AD build.
# $DomainName - this is the full name of the domain that you will be adding the DC
# $DomainUser - this account must have the Domain Admin role
# $EncryptedPass - the encrypted password
# $Credential - the encrypted domain
# $DnsServer - this is the private IP address of the Primary Domain Controller
Try {
Start-Transcript -Path "C:\DomainJoin\Stage3.txt" -Force
$Password="Password!!"
$DomainName="CMGSOL.corp"
$DomainUser="cmgsol\administrator"
$EncryptedPass = ConvertTo-SecureString $Password -AsPlainText -Force
$Credential = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList $DomainUser, $EncryptedPass
# Enter the address for the domain controller.
$DnsServer = '10.10.0.1'
#Set the Administrator Password and activate the Domain Admin Account
net user Administrator $Password /logonpasswordchg:no /active:yes
#####
# Create the Second Domain Controller
#
#####
Install-WindowsFeature NET-Framework-Core
Install-WindowsFeature AD-Domain-Services
Install-WindowsFeature RSAT-ADDS
Install-WindowsFeature RSAT-DNS-Server
Set-DnsClientServerAddress -InterfaceAlias Ethernet -ServerAddresses $DnsServer
Install-ADDSDomainController -InstallDns -Credential $Credential -DomainName $DomainName -SafeModeAdministratorPassword
$EncryptedPass -Force -NoRebootOnCompletion
} Catch {
Write-Host $_
} Finally {
Stop-Transcript

```



```

}
start-sleep -s 300
Restart-Computer -ComputerName "localhost" -Force

```

## Appendix C: AddComputer.ps1

```

#ps1_sysnative
#####
# Title: AddComputer.ps1
# Version & Date: v1 31 Oct 2018
# Updated: v2 31 Mar 2022
# Creator: john.s.parker@oracle.com
# Warning: This script is a representation of how to use PowerShell to add a computer to an Active Directory Domain.
#         This script creates and uses the domain administrator account there are potential for mistakes and destructive actions.
#         USE AT YOUR OWN RISK!!
# Source:
# From https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.management/add-computer?view=powershell-5.1#examples
# Variables for this script
# $NewComputerName - this is the name of the new computer that you want to add to your domain
#
## Run as Administrator on a domain computer.
$NewComputerName = "WS16CN001"
New-ADComputer -Name $NewComputerName -AccountPassword (ConvertTo-SecureString -String 'TempJoinPA$$' -AsPlainText -Force)

```

## Appendix D: NewComputer.ps1

```

#ps1_sysnative
#####
# Title: newcomputer.ps1
# Version & Date: v1 31 Oct 2018
# Udated: v2 31 Mar 2022
# Creator: lawrence.gabriel@oracle.com & john.s.parker@oracle.com
# Warning: This script is a representation of how to use PowerShell to add a new computer to an Active Directory Domain
#         Warning there are potential for mistakes and destructive actions. USE AT YOUR OWN RISK!!
#         This is the fourth script in the Active Directory Series that will join a computer to your new Active Directory Domain.
#         This script will join the newly created host to an Active Directory Domain.
#
# Variables for this script
# $DnsServer - this is the private IP address of the Primary Domain Controller
# $DnsServer2 - this is the private IP address of the Secondary Domain Controller
# $DomainToJoin - this is the full name of the domain you want to join.
# $JoinCred - this will be the encrypted credential
#
Try {
Start-Transcript -Path "C:\DomainJoin\Stage4.txt" -Force
$DnsServer = '192.168.0.1'
$DnsServer2 = '192.168.0.2'
$DomainToJoin = 'cesa.corp'
#####
# Sets the DNS to the DC.
#####
Set-DnsClientServerAddress -InterfaceAlias Ethernet -ServerAddresses ($DnsServer, $DnsServer2)
#####
# Build the one time use password
#####
$JoinCred = New-Object pscredential -ArgumentList ([pscustomobject]@{
    UserName = $null
    Password = (ConvertTo-SecureString -String 'TempJoinPA$$' -AsPlainText -Force)[0]
})
Add-Computer -Domain $DomainToJoin -Options UnsecuredJoin,PasswordPass -Credential $JoinCred
} Catch {
Write-Host $_
} Finally {
}
Stop-Transcript
}
#####
#
# This wait is to ensure that the Add-Computer command finishes before the restart.
#
#####

```

```
start-sleep -s 300
Restart-Computer -ComputerName "localhost" -Force
```

---

## Connect with us

Call **+1.800.ORACLE1** or visit **oracle.com**. Outside North America, find local offices at **oracle.com/contact**.

 [blogs.oracle.com](https://blogs.oracle.com)

 [facebook.com/oracle](https://facebook.com/oracle)

 [twitter.com/oracle](https://twitter.com/oracle)

---

Copyright © 2022, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0120