Kapitel 3

Erste Schritte mit Windows Server 2016

In diesem Kapitel:

Nachdem Sie in den beiden vorherigen Kapiteln bereits einige grundlegenden Informationen zu den Neuerungen und zur Installation von Windows Server 2016 erhalten haben, erfahren Sie in diesem Kapitel, welche ersten Schritte Sie zur Verwaltung von Windows Server 2016 durchführen müssen.

Erste Schritte nach der Installation

Während der Installation legt Windows Server 2016 automatisch einen Namen für den Server fest, der nachträglich angepasst werden sollte. Wie Sie dabei vorgehen, lesen Sie in Kapitel 2. Viele Aufgaben, die zur Grundkonfiguration des Servers gehören, nehmen Sie direkt im Server-Manager vor. Dazu klicken Sie auf *Lokaler Server*. Im mittleren Bereich sehen Sie die verschiedenen Aufgaben, deren Assistenten Sie über einen Klick auf den entsprechenden Link erreichen.

Windows Server 2016 mit Windows 10 verwalten

Um Windows Server 2016 mit Windows 10 zu verwalten, bietet Microsoft die Remoteserver-Verwaltungstools (Remote Server Administration Tools, RSAT) zum Download an (*http://tinyurl.com/jmrdeea*). Mit den Tools installieren Sie auf einer Arbeitsstation mit Windows 10 alle Programme, die zur Verwaltung von Windows Server 2016 notwendig sind. Mit den Tools verwalten Sie ebenfalls die Serverdienste in Windows Server 2012/2012 R2. Auch Container und Nano-Server können Sie über diesen Weg verwalten. Sie brauchen dazu aber Windows 10 Enterprise-Version 1607 oder neuer.

Neben den verschiedenen Verwaltungstools der Serverrollen integriert der Installations-Assistent von RSAT auch den Server-Manager von Windows Server 2016 in Windows 10. Über den Server-Manager binden Sie die verschiedenen Server im Netzwerk an, auf denen Windows Server 2016 installiert ist. Sie können mit dem Server-Manager auf diesem Weg auch von Windows 10-Arbeitsstationen aus Serverrollen auf Servern installieren. Auch im Server-Manager von Windows Server 2016 können Sie andere Server mit Windows Server 2016 im Netzwerk verwalten.

Die Remoteserver-Verwaltungstools für Windows 10 umfassen den Server-Manager, die Verwaltungstools der Serverrollen und Features von Windows Server 2016, PowerShell-Cmdlets und Befehlszeilentools für die Verwaltung von Rollen und Features. Einige Tools lassen sich zur Verwaltung von Rollen und Features in Windows Server 2008 R2 und Windows Server 2012/2012 R2 nutzen.

Die Remoteserver-Verwaltungstools können Sie zwar auch in der kleinsten Version Windows 10 installieren, allerdings bietet nur die Enterprise-Version alle Funktionen. Sie können die Remoteserver-Verwaltungstools für Windows 10 nur auf Computern installieren, auf denen Windows 10 installiert ist, nicht auf Rechnern mit der Server-Version von Windows.

Remoteserver-Verwaltungstools installieren

Die Remoteserver-Verwaltungstools laden Sie als *.msu*-Datei direkt vom Microsoft-Downloadcenter herunter. Der Download steht als 64-Bit- und als 32-Bit-Version zur Verfügung. Bei der Installation wählen Sie keine Verwaltungstools aus, sondern installieren lediglich die Tools als Update in Windows 10.

Windows 10 installiert RSAT wie jedes andere Update, das heißt, die Installation lässt sich auch skripten. Entfernen Sie vorher alle älteren Versionen der Verwaltungstools oder Remoteserver-Verwaltungstools, selbst früherer Vorabversionen sowie Versionen der Tools für verschiedene Sprachen.

Wenn Sie ein Upgrade von Windows 7/8.1 auf Windows 10 durchgeführt haben, müssen Sie die Remoteserver-Verwaltungstools für Windows 10 installieren, Sie können nicht die alten Versionen für Windows 7/8.1 parallel betreiben. Die Remoteserver-Verwaltungstools für Windows 10 unterstützen die Remoteverwaltung von Servern mit einer Core-Installation und teilweise auch die Server Core-Installationen von Windows Server 2008 R2 oder Windows Server 2008.

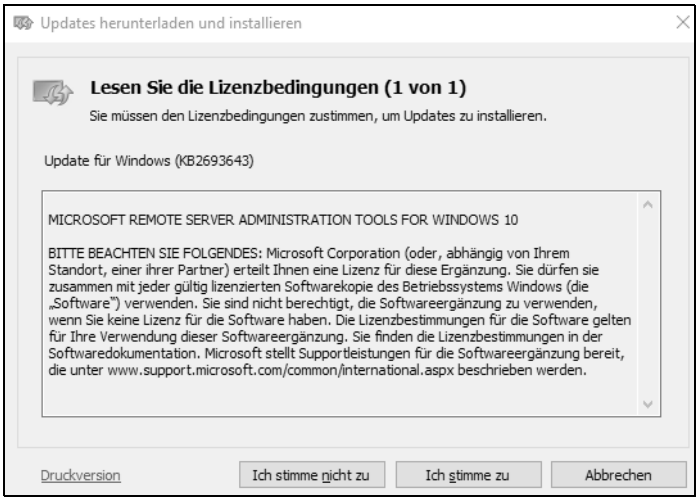

Abbildung 3.1: Die Remoteserver-Verwaltungstools stehen als Update zur Verfügung.

Nach der Installation finden Sie die Remoteserver-Verwaltungstools im Startmenü. Im Gegensatz zu Windows 7 sind alle Verwaltungstools nach der Installation bereits aktiv. Wollen Sie nicht alle Verwaltungstools nutzen, können Sie einzelne davon deaktivieren. Dazu tippen Sie »optionalfeatures« im Suchfeld des Startmenüs ein und suchen im Dialogfeld *Windows-Features* den Abschnitt *Remoteserver Administration Tools*. Hier aktivieren oder deaktivieren Sie einzelne Verwaltungstools. Zur Installation müssen Sie nur das jeweilige Kontrollkästchen aktivieren, eine weitere Installation ist nicht notwendig. Wollen Sie die Tools komplett deinstallieren, gehen Sie folgendermaßen vor:

1. Rufen Sie über das Suchfeld der Startseite *Appwiz.cpl* auf.
2. Klicken Sie auf *Installierte Updates anzeigen*.
3. Klicken Sie mit der rechten Maustaste auf *Update für Microsoft Windows (KB2693643)* und dann auf *Deinstallieren*.
4. Bestätigen Sie die Deinstallation des Updates mit *Ja*.

Remoteverwaltung mit dem Server-Manager

Das Erste, was nach der Installation von Windows Server 2016 auffällt, ist die im Vergleich zu Windows Server 2008 R2 überarbeitete Version des Server-Managers. Im Vergleich zu Windows Server 2012/2012 R2 sind keine Neuerungen zu sehen. Der Server-Manager bietet aber im Vergleich zu Windows Server 2008 R2 nicht nur eine neue Oberfläche, sondern auch mehr Funktionen. So ist es in der neuen Version möglich, Serverrollen und Features über das Netzwerk auf anderen Servern zu installieren.

Die Server im Netzwerk lassen sich zentral im Server-Manager verwalten. Klicken Sie im Server-Manager auf *Dashboard*, können Sie über das Menü *Ansicht* die Willkommen-Kachel ausblenden und gewinnen wertvollen Platz zur Verwaltung von Servern. Über die Programmgruppe *Verwalten* erstellen Sie eigene Servergruppen.

Dazu gruppiert der Server-Manager die verschiedenen Serverfunktionen zur besseren Verwaltung. Alle installierten Serverrollen werden im Server-Manager automatisch gruppiert. Verwaltungswerkzeuge zeigt der Server-Manager direkt über das Menü *Tools* an. Hierüber lassen sich alle wichtigen Werkzeuge starten. So stört die neue Oberfläche nicht, da alle

Verwaltungsaufgaben zentral im Server-Manager stattfinden. Diese Funktionen sind nach der Installation von RSAT außerdem in Windows 10 verfügbar.

Um im Server-Manager in Windows Server 2016 und Windows 10 weitere Server anzubinden, klicken Sie auf *Verwalten* und dann auf *Server hinzufügen*. Im Fenster können Sie anschließend nach Servern suchen, um sie im lokalen Server-Manager zu verwalten. Auf diesem Weg erstellen Sie eigene Servergruppen, die Sie im Server-Manager zusammenfassen. Von diesen Gruppen können Sie dann Ereignismeldungen anzeigen lassen. Über diesen Weg binden Sie Server mit Windows Server 2016 in allen Editionen, aber auch Windows Server 2012/2012 R2 an.

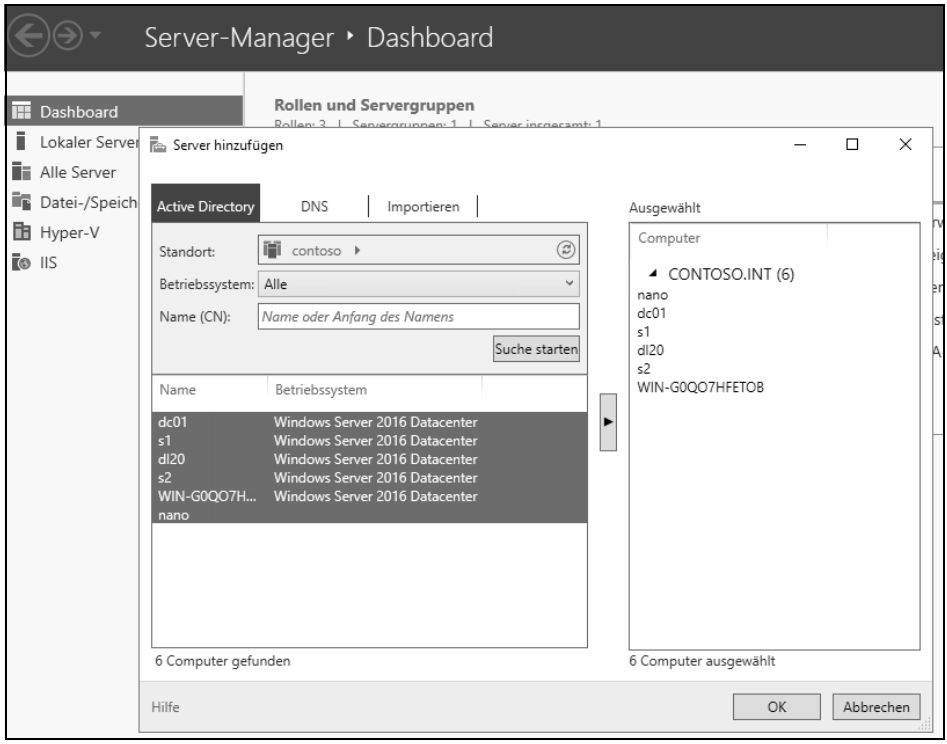

Abbildung 3.2: Verwalten von zusätzlichen Servern im Server-Manager

Um auf Servern im Netzwerk über den Server-Manager remote Rollen oder Features zu installieren, ist eine vorherige Anbindung notwendig. Im Assistenten zum Hinzufügen von zusätzlichen Rollen erscheint ein Fenster, über das Sie den Server auswählen können, auf dem Sie eine neue Rolle oder ein neues Feature installieren wollen. Dazu klicken Sie auf *Verwalten/Rollen und Features* hinzufügen.

Hier fällt eine weitere Neuerung im Vergleich zu Windows Server 2008 R2 auf. In Windows Server 2016 sind die Assistenten zum Hinzufügen von Rollen und Features zusammengefasst. Das ist bereits seit Windows Server 2012 so. Das heißt, Sie können über einen einzelnen Assistenten mehrere Serverrollen und Features gemeinsam und gleichzeitig installieren. Dies erspart unnötige Neustarts und Installationen, da sämtliche Aufgaben in einem Arbeitsschritt durchgeführt werden. Im Assistenten lassen sich aber nicht nur

physische Server im Netzwerk auswählen, um Serverrollen zu installieren, sondern auch virtuelle Festplatten auf Hyper-V-Hosts.

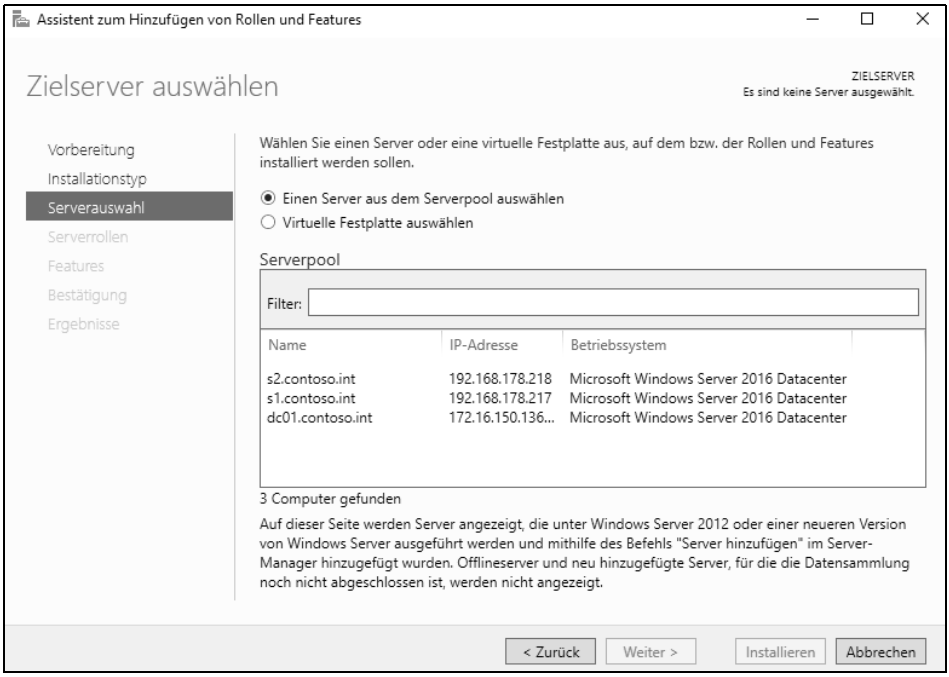

Abbildung 3.3: Auswählen des Zielservers zur Installation von Serverrollen

Beim Abschluss der Installation von Serverrollen und Features erhalten Sie eine Zusammenfassung angezeigt und die Möglichkeit geboten, die Konfiguration in *.xml*-Dateien zu exportieren. Mit dieser Datei können Sie dann die gleichen Rollen oder Features auf einem anderen Server installieren. Zusätzlich haben Sie die Möglichkeit, einen alternativen Pfad zu den Installationsdateien von Windows Server 2016 anzugeben. Hier sollten Sie auch die Option zum automatischen Neustart aktivieren.

In diesem Fall starten die Server automatisch neu, falls dies notwendig ist. Vor allem, wenn Sie Installationen von Serverrollen über das Netzwerk oder über eine Remotedesktopverbindung ausführen, ist dies sinnvoll, da viele Rollen (beispielsweise die Installation von Hyper-V) die Netzwerkverbindung kappen können. Während der Assistent die Aufgaben durchführt, müssen Sie das Fenster nicht geöffnet lassen, sondern können es nach dem Start der Installation schließen.

Überall im Server-Manager lassen sich die anderen Server im Netzwerk schnell und einfach integrieren sowie verwalten. Über das Kontextmenü von Servern können Sie Server über das Netzwerk remote neu starten lassen, eine PowerShell-Sitzung auf dem Server starten oder eine Remotedesktopverbindung öffnen. Auch die Installation von Rollen und Features über das Netzwerk ist mit dem Kontextmenü möglich.

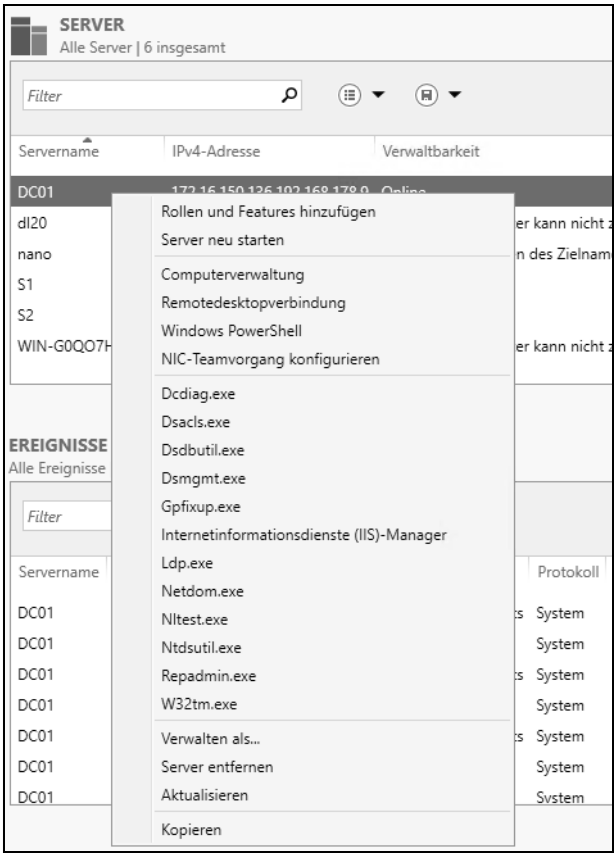

Abbildung 3.4: Über das Kontextmenü von Servern lassen sich die Verwaltungswerkzeuge von Windows Server 2016 auch in Windows 10 starten.

Im Server-Manager sehen Administratoren am Wartungscentersymbol im oberen Bereich, ob Fehler auf einem angebundenen Server vorliegen oder Maßnahmen zur Verwaltung notwendig sind. Sie können sich über diesen Weg in Windows 10 auch gesammelt alle Fehlermeldungen aller Server anzeigen lassen.

Klicken Sie in der Ansicht *Alle Server* auf einen Server im oberen Bereich, sehen Sie unten wichtige Fehlermeldungen der Ereignisanzeige. Im oberen Bereich ist außerdem zu sehen, ob die entsprechenden Server online sind und ob Windows Server 2016 aktiviert ist.

Nach der Installation von Windows Server 2016 sollten Sie im Server-Manager über das Kontextmenü der Server den Befehl *Leistungsindikatoren starten* ausführen, damit der Server über das Netzwerk überwachbar ist und die neuen Best Practices Analyzer funktionieren und Daten abrufen können. Über das Kontextmenü der Server können Sie sich außerdem mit einem anderen Benutzernamen am Server anmelden, um diesen zu administrieren.

Tipp

Wenn Sie auf einem Core-Server nur einen schwarzen Bildschirm sehen, ist die Eingabeaufforderung geschlossen. Um diese zu öffnen, drücken Sie Strg+Alt+Entf und starten den Task-Manager. Mit *Mehr Details* und Eingabe von »cmd« über *Datei/Neuen Task ausführen* starten Sie die Eingabeaufforderung neu.

Um das Verwaltungsprogramm von Core-Servern aufzurufen, geben Sie *Sconfig* ein. Das Befehlszeilentool *Sconfig* steht in Windows Server 2016 auch auf Servern mit grafischer Benutzeroberfläche zur Verfügung. Auf diesem Weg können Sie zum Beispiel in Fernwartungen Einstellungen vornehmen, wenn die Verbindung für grafische Werkzeuge zu langsam ist.

Core-Server und Hyper-V Server 2016 verwalten

Core-Server hat Microsoft mit Windows Server 2008 R2 eingeführt und mit Windows Server 2012/2012 R2 verbessert. In Windows Server 2016 bieten Core-Server ähnliche Funktionen wie in Windows Server 2012 R2. Den Servern fehlt die grafische Oberfläche. Sie verwalten sie mit der Eingabeaufforderung, der PowerShell oder über das Netzwerk von anderen Servern oder auch Windows 10-Arbeitsstationen. Das Gleiche funktioniert ebenfalls für den neuen Hyper-V-Server 2016. Der Hyper-V Server 2016 ist im Grunde genommen ein Core-Server mit automatisch installierter Hyper-V-Rolle.

Hinweis

Core-Server lassen sich in Windows Server 2016 nicht auf Server mit grafischer Oberfläche aktualisieren und umgekehrt lässt sich die grafische Oberfläche nach der Einrichtung nicht deinstallieren.

Haben Sie aber die Remoteserver-Verwaltungstools (Remote Server Administration Tools, RSAT) in Windows 10 installiert, können Sie diese auch von einer Windows 10-Arbeitsstation aus verwenden, ohne dass auf dem Core-Server eine grafische Oberfläche zur Verfügung steht.

Sie können Hyper-V Server 2016 mit dem Hyper-V Manager in Windows 10 verwalten, auch ohne RSAT zu nutzen. Wichtig für die Verwaltung von Core-Servern oder Hyper-V Server 2016 über das Netzwerk sind noch die Punkte 4 und 7 in Sconfig. Hierüber aktivieren Sie die Remoteverwaltung mit Tools wie den Hyper-V-Manager. Durch Aktivierung des Remotedesktops lässt sich Hyper-V Server 2016 auch darüber verwalten. Wie Sie dabei vorgehen, lesen Sie in Kapitel 2.

Haben Sie sich mit einem Core-Server verbunden und versehentlich die Eingabeaufforderung geschlossen, drücken Sie die Tastenkombination Strg+Alt+Entf und starten den Task-Manager. Klicken Sie danach auf *Mehr Details* und dann auf *Datei/Neuen Task ausführen*. Tippen Sie »cmd« ein, um die Eingabeaufforderung erneut zu öffnen.

Haben Sie einen Core-Server installiert, legen Sie zunächst die IP-Adresse fest, konfigurieren den DNS-Server, ändern den Namen und nehmen den Server in die Active Directory-Domäne auf. Aktivieren Sie noch die Remoteverwaltung, können Sie den Server mit grafischen Verwaltungstools verwalten, wie in den ersten Abschnitten in diesem Kapitel behandelt.

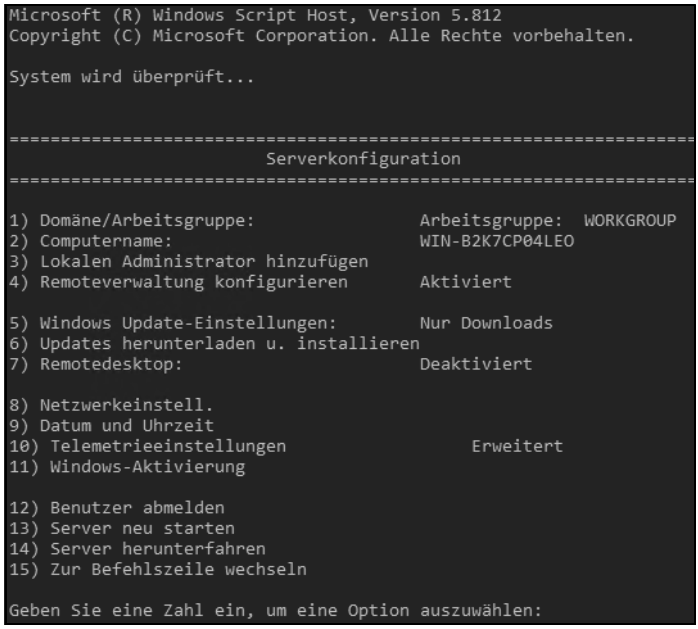

Abbildung 3.5: Die Verwaltung von Core-Servern erfolgt unter anderem mit Sconfig.

Um Core-Server zu verwalten, rufen Sie zunächst in der Eingabeaufforderung den Befehl *Sconfig* auf. Zur Konfiguration der Netzwerkeinstellungen wählen Sie den Menüpunkt *8) Netzwerkeinstellungen*:

1. Wählen Sie die Nummer des Adapters aus.
2. Wählen Sie *1) Adresse der Netzwerkkarte festlegen* aus, um die Adresse zu ändern.
3. Geben Sie *S* ein, um eine statische IP-Adresse zu konfigurieren.
4. Geben Sie die statische IP-Adresse ein und danach die Subnetzmaske.

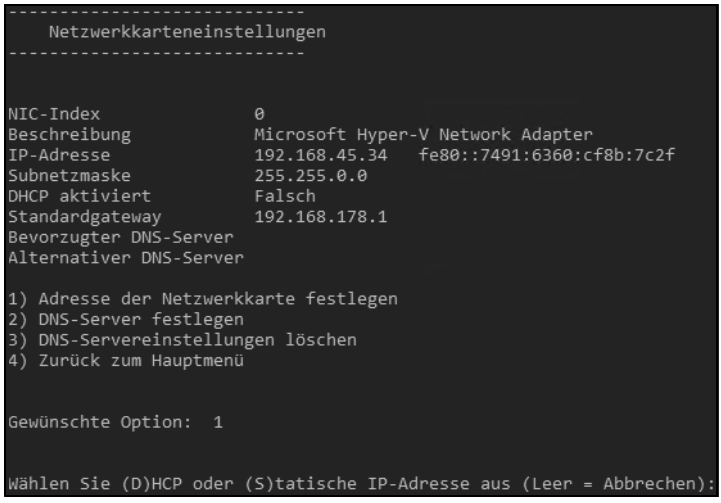

Abbildung 3.6: Festlegen einer statischen IP-Adresse für einen Core-Server

5. Anschließend tragen Sie über den Menüpunkt *2) DNS-Server festlegen* einen DNS-Server ein, der die Active Directory-Domäne auflösen kann.
6. Im Hauptmenü zurück nehmen Sie den Server mit dem Punkt *1) Domäne/Arbeitsgruppe* in die Domäne auf und ändern den Servernamen. Anschließend starten Sie den Server neu.
7. Über die Menüpunkte 4 und 7 im Sconfig-Hauptmenü aktivieren Sie die Verwaltung des Remotedesktops und die Remoteverwaltung über grafische Tools wie den Server-Manager.

Die Verwaltung eines Core-Servers läuft hauptsächlich über die Eingabeaufforderung oder PowerShell ab beziehungsweise mit Verwaltungstools über das Netzwerk.

Tipp

Mit dem Befehl *Start cmd /separate* öffnen Sie ein paralleles Fenster der Eingabeaufforderung, wenn Sie zwei Fenster benötigen. Wird das eine Fenster geschlossen, lässt sich über den Task-Manager durch Erstellen eines neuen Tasks mit dem Befehl *Cmd* ein neues Fenster starten, aber mit einem zweiten Fenster ersparen Sie sich diesen Aufwand und können bei der Arbeit mit einem Skript parallel mit einer zweiten Oberfläche arbeiten.

Alle Tools, die eine grafische Oberfläche verwenden oder den Explorer benötigen, funktionieren auf einem Core-Server nicht. Aus diesem Grund werden auch keine Meldungen angezeigt, wenn neue Updates zur Verfügung stehen oder das Kennwort eines Benutzers abgelaufen ist. Einige Fenster funktionieren außerdem auf einem Core-Server. So kann zum Beispiel der Editor (Notepad) verwendet werden, um Skripts oder Dateien zu bearbeiten. Mit Notepad können Sie das Dateisystem durchsuchen und Skripts bearbeiten. Der Task-Manager steht ebenfalls zur Verfügung.

Um das lokale Administratorkennwort eines Servers anzupassen, gehen Sie folgendermaßen vor:

1. Rufen Sie in der Eingabeaufforderung den Befehl *Net user administrator ** auf. Durch die Eingabe des Platzhalters * wird das eingegebene Kennwort nicht im Klartext angezeigt.
2. Geben Sie das neue Kennwort ein und bestätigen Sie.
3. Geben Sie das Kennwort noch mal ein und bestätigen Sie erneut.

Sie können auch Einstellungen des Servers in der Eingabeaufforderung anpassen. Das Kennwort des angemeldeten Benutzers ändern Sie über die Tastenkombination Strg + Alt + Entf. Die PowerShell ist in Core-Installationen automatisch aktiviert. Daher verwenden Sie zur Konfiguration der IP-Einstellungen nicht mehr das Befehlszeilentool *Netsh*, sondern besser die Cmdlets *New-NetIPAddress* und *Get-NetIPConfiguration*.

Ein Beispiel für die Einrichtung ist:

New-NetIPAddress -InterfaceIndex 12 -IPAddress 192.161078.2 -PrefixLength 24 -DefaultGateway 192.1610710

Die DNS-Server tragen Sie ein mit:

Set-DnsClientServerAddress -InterfaceIndex 12 -ServerAddresses 192.1610784

Mehrere DNS-Server trennen Sie jeweils mit einem Komma. Das folgende Cmdlet wechselt zu DHCP:

Set-DnsClientServerAddress -InterfaceIndex 12 -ResetServer

Achten Sie darauf, jeweils die korrekte Indexnummer für den Netzwerkadapter zu verwenden. Diesen erhalten Sie mit dem Aufruf des Cmdlets *Get-NetIPConfiguration*.

Einer Windows-Domäne treten Sie mit *Add-Computer* bei. Um der lokalen Administratorengruppe ein Domänenkonto hinzuzufügen, verwenden Sie den Befehl *Net localgroup administratoren /add <Domäne>\<Benutzername>*. Mit dem Befehl *Net localgroup administratoren* können Sie sich alle Gruppenmitglieder anzeigen lassen. Die Aufnahme funktioniert auch über Sconfig, geht aber mit der Eingabeaufforderung schneller.

Mit dem Befehl *Net localgroup* können Sie sich alle lokalen Gruppen auf dem Server anzeigen lassen. So können Sie mit diesem Befehl schnell feststellen, welche Gruppen es gibt und welche Benutzerkonten enthalten sind. Außerdem lassen sich neue Benutzerkonten hinzufügen. Sie können die Benutzerverwaltung auch über die grafische Oberfläche von einem anderen Server aus durchführen, wenn Sie die Remoteverwaltung auf dem Server aktiviert haben. Mit dem Befehl *Net localgroup administratoren /delete <Domäne>\<Benutzername>* entfernen Sie ein Benutzerkonto wieder aus der Gruppe.

Den Namen von Servern ändern Sie mit *Rename-Computer*. Der Aufruf von *Set-Date* ändert die Zeitzone, und die Spracheinstellungen ändern Sie mit *Control intl.cpl*.

Tipp Installieren Sie Windows-Installer-Pakete auf einem Core-Server, verwenden Sie beim Aufruf die Option */qb*.

Die Computerverwaltung starten Sie zum Beispiel über das Snap-In *Active Directory-Benutzer und -Computer*. Klicken Sie den Core-Server in der Konsole mit der rechten Maustaste an und wählen Sie im Kontextmenü den Eintrag *Verwalten*. Anschließend kann der Server über eine grafische Oberfläche konfiguriert werden. Über diesen Weg lassen sich zum Beispiel wesentlich einfacher Freigaben und Systemdienste verwalten als über die Eingabeaufforderung des Core-Servers.

Hardware und Treiber auf Core-Servern installieren

Installieren Sie neue Hardware, können Sie die grafische Oberfläche oder die Eingabeaufforderung verwenden. Auf Core-Servern bleibt Ihnen keine andere Wahl, als die Eingabeaufforderung zu verwenden. Haben Sie die neue Hardware mit dem Server verbunden, wird sie durch die Plug&Play-Funktion automatisch erkannt und der Treiber installiert. Das gilt auch bei Core-Servern. Allerdings muss in diesem Fall der Treiber in Windows Server 2016 integriert sein. Ist er das nicht und müssen Sie den Treiber manuell nachinstallieren, gehen Sie folgendermaßen vor:

1. Entpacken Sie die Treiberdateien und kopieren Sie sie in einen Ordner auf dem Server.
2. Geben Sie den Befehl *Pnputil -i -a <*.inf-Datei des Treibers>* ein. Mit diesem neuen Tool können Treiber in Windows Server 2016 hinzugefügt und entfernt werden.
   - Über den Befehl *Sc query type= driver* können Sie sich alle installierten Treiber auf einem Server anzeigen lassen (achten Sie auf das Leerzeichen nach dem Gleichheitszeichen).

– Mit dem Befehl *Sc delete <Treibername>* können Sie den Treiber entfernen, den Sie sich zuvor über den Befehl *Sc query type= driver* anzeigen lassen können.

Für die Anbindung an iSCSI-Targets (siehe Kapitel 5) steht auf Core-Servern eine grafische Oberfläche zur Verfügung. Diese starten Sie durch Eingabe des Befehls *Iscsicpl*. Für die Anbindung von Core-Servern an iSCSI-Targets steht auch der Befehl *Iscsicli* zur Verfügung. Über *Iscsicli /?* erhalten Sie eine ausführliche Hilfe zum Befehl (siehe Kapitel 5).

Windows Updates auf Core-Servern steuern

Um Windows-Updates zu steuern, verwenden Sie auf Core-Servern ebenfalls Sconfig. Mehr zu diesem Thema lesen Sie in Kapitel 37.

Um eine sofortige Installation von Updates durchzuführen, geben Sie den Befehl *Wuauclt /detectnow* ein. Die installierten Updates lassen sich durch den Aufruf von *Systeminfo* oder *Wmic qfe list* anzeigen.

Erweiterte Startoptionen nutzen

Die erweiterten Startoptionen bieten Möglichkeiten zur Reparatur des Servers. Wir gehen in Kapitel 35 noch ausführlicher auf dieses Thema ein. Die Optionen lassen sich zum Beispiel aufrufen, wenn der Server beim Starten einige Male abstürzt. Hier stehen verschiedene Möglichkeiten zur Verfügung:

- **Computer reparieren** – Startet die Reparatur des Betriebssystems in der Recovery-Oberfläche.
- **Abgesicherter Modus** – Startet Windows mit den mindestens erforderlichen Treibern und Diensten.
- **Abgesicherter Modus mit Netzwerktreibern** – Startet Windows im abgesicherten Modus zusammen mit den für den Zugriff auf das Internet oder auf andere Computer im Netzwerk erforderlichen Netzwerktreibern und -diensten.
- **Abgesicherter Modus mit Eingabeaufforderung** – Startet Windows im abgesicherten Modus mit einem Eingabeaufforderungsfenster anstelle der normalen Windows-Benutzeroberfläche.
- **Startprotokollierung aktivieren** – Erstellt die Datei *Ntbtlog.txt*, in der alle Treiber aufgelistet werden, die beim Starten installiert werden und für die erweiterte Problembehandlung nützlich sein können.
- **Videomodus mit niedriger Auflösung aktivieren** – Startet Windows mithilfe des aktuellen Videotreibers und mit niedrigen Einstellungen für Auflösung und Aktualisierungsrate. In diesem Modus können Sie die Anzeigeeinstellungen zurücksetzen.
- **Letzte als funktionierend bekannte Konfiguration** – Startet Windows mit der letzten funktionsfähigen Registrierungs- und Treiberkonfiguration.
- **Debugmodus** – Startet Windows in einem erweiterten Problembehandlungsmodus.
- **Automatischen Neustart bei Systemfehler deaktivieren** – Verhindert, dass Windows nach einem durch einen eigenen Fehler verursachten Absturz automatisch neu gestartet wird. Wählen Sie diese Option nur aus, wenn Windows in einer Schleife festgefahren ist, die aus Absturz, Neustart und erneutem Absturz besteht.
- **Erzwingen der Treibersignatur deaktivieren** – Ermöglicht, dass Treiber mit ungültigen Signaturen installiert werden.

- **Frühen Start des Treibers der Antischadsoftware deaktivieren** – In Windows Server 2016 startet der installierte Virenscanner wesentlich früher als in Windows Server 2008 R2. Das kann zu Problemen führen, wenn der Computer nicht mehr startet. Hier deaktivieren Sie diesen Schutz.

Windows Remote Management (WinRM) aktivieren (auch für Nano-Server)

Über Windows Remote Management (WinRM) lassen sich Cmdlets remote auf Nano-Servern, aber auch auf herkömmlichen Windows-Servern ausführen. Damit das funktioniert, muss auf dem Server, der eine Verbindung zum Nano-Server aufbaut, WinRM konfiguriert werden. Dazu müssen die folgenden Befehle in einer Eingabeaufforderung mit administrativen Rechten eingegeben werden:

Winrm quickconfig

Winrm set winrm/config/client @{TrustedHosts=""}*

Chcp 65001

Anschließend lässt sich in der Eingabeaufforderung eine Verbindung aufbauen:

Winrs -r:<IP-Adresse des Nano-Servers> -u:Administrator -p:<Kennwort > <Befehl, zum Beispiel Ipconfig>

Der Befehl wird in diesem Fall auf dem Nano-Server ausgeführt. So lassen sich außerdem Skripts für die Ausführung von Befehlen schreiben. WMI steht aber auch in der PowerShell zur Verfügung, wenn Administratoren eine Verbindung zum Nano-Server aufbauen.

Wollen Sie auf einem Nano-Server Daten von Festplatten auslesen, stehen verschiedene Möglichkeiten zur Verfügung. Der einfachste Weg ist die Verwendung des Cmdlets *Get-PhysicalDisk*. Dieses Cmdlet steht auch bei herkömmlichen Servern zur Verfügung und lässt sich ebenfalls lokal einsetzen. Die PowerShell zeigt für eine Liste der Laufwerke an, ob diese Mitglied eines Speicherpools sein können oder sind, wie der Status des Laufwerks ist, und dessen maximale Größe. Noch mehr Informationen erhalten Sie mit *Get-PhysicalDisk |fl*.

Zusammenfassung

In diesem Kapitel haben wir Ihnen gezeigt, wie Sie mit der neuen Oberfläche in Windows Server 2016 umgehen. Wir sind darauf eingegangen, wie Sie Server im Netzwerk verwalten und mit dem Server-Manager in Windows Server 2016 umgehen.

Auch die Verwaltung und Einrichtung von Core-Servern sowie die Verwaltung von Windows Server 2016 mit den Remoteserver-Verwaltungstools in Windows war Bestandteil des Kapitels.

Im nächsten Kapitel erfahren Sie, wie Serverrollen und Features in Windows Server 2016 installiert werden. Auch hier hat sich einiges im Vergleich zu Windows Server 2008 R2 verändert.

Kapitel 26

Windows Server-Container, Docker und Hyper-V-Container

In diesem Kapitel:

Neben Nano-Servern gehören Windows Server-Container als Docker-Implementation zu den wichtigsten Neuerungen in Windows Server 2016. Windows Server-Container lassen sich auch auf Nano-Servern ausführen und Nano-Server lassen sich wiederum als Cluster betreiben. Zusätzlich können Container sowohl in virtuellen Umgebungen als auch in virtuellen Clustern hochverfügbar zur Verfügung gestellt werden.

Container lassen sich somit auf allen Arten von Windows-Servern betreiben, also auf Nano-, Core- und Windows-Servern mit grafischer Oberfläche. Außerdem steht die Container-Technologie in Windows 10 Professional und Enterprise ab Version 1607 (Anniversary Update) zur Verfügung. Entsprechend können Administratoren oder Entwickler mit Windows 10 ebenfalls Container erstellen und diese auf Container-Hosts mit Windows Server 2016 übertragen. Hier kann auch auf den Docker-Hub in der Cloud gesetzt werden, um die Container-Images über das Internet und die Cloud zu übertragen.

Die Grundlagen zu Containern und Docker

Windows Server-Container ermöglichen den Betrieb von Cloudanwendungen oder Webdiensten in einer sicheren und einfach zu erstellenden Umgebung. Alles, was Sie benötigen, ist ein Container-Host auf Basis von Windows Server 2016. Dabei kann es sich um einen physischen Server handeln, eine virtuelle Maschine (VM) oder einen virtuellen Computer in Microsoft Azure.

Innerhalb des Container-Hosts, der zum Beispiel auf Basis eines Nano- oder Core-Servers mit Windows Server 2016 zur Verfügung gestellt wird, verwalten Sie die Images für Container und die Container selbst. Die Verwaltung findet vor allem über die PowerShell oder die Eingabeaufforderung statt. Auch die Container werden hierüber verwaltet. Der Verbindungsaufbau zum Container-Host kann über eine RDP-Sitzung erfolgen.

Container im Vergleich zu virtuellen Servern

Die Windows Server-Container sowie deren Erweiterung Hyper-V-Container basieren auf der Plattform Docker (*https://www.docker.com*). Microsoft arbeitet eng mit den Entwicklern von Docker zusammen, um eine optimale Integration von Docker zu gewährleisten. Die Verwaltung von Docker können Sie mit dem Docker-Client oder in der PowerShell vornehmen. Sie können Container auch mit System Center 2016 verwalten.

Virtualisieren Unternehmen Server auf herkömmlichen Technologien, gibt es einige Nachteile. Ein Nachteil besteht zum Beispiel darin, dass die Betriebssysteme in den virtuellen Servern eine Grundlast verursachen und damit Ressourcen verbrauchen und Sicherheitslücken darstellen.

Das Betriebssystem in Docker-Containern und die notwendigen Ressourcen sind auf dem Container-Host zusammengefasst. Startet ein Container, muss er nicht das komplette Betriebssystem booten, Bibliotheken laden und Ressourcen für das eigene Betriebssystem zur Verfügung stellen. Stattdessen nutzen Container nur Teile des Betriebssystems auf dem Container-Host. Die Vorteile dabei sind eine geringere Auslastung der Server und mehr Sicherheit. Der gestartete Container betrachtet die lokale Festplatte wie eine Kopie des Betriebssystems, inklusive Arbeitsspeicher, Dateien und andere Ressourcen.

Virtuelle Anwendungen sind kleiner als virtuelle Server, benötigen weniger Ressourcen und sind gleichzeitig sicherer, da die meisten Angriffspunkte fehlen. Außerdem lassen sich wesentlich mehr virtuelle Anwendungen auf einem Virtualisierungs-Host betreiben als herkömmliche virtuelle Server.

Windows Server-Container unterstützen zahlreiche Programmiersprachen und -Umgebungen. Entwickler können unter anderem.NET, ASP.NET, PowerShell, Python, Ruby on Rails, Java und viele andere Umgebungen nutzen. Der Container-Host auf Basis von Windows Server 2016 steuert, welche und wie viele Ressourcen des Hosts ein Container nutzen darf, ohne die anderen Container oder den Host zu beeinträchtigen.

Das Container-Feature installieren

Um Container zu nutzen, müssen Sie das Container-Feature installieren. Dabei spielt es zunächst keine Rolle, ob es sich um einen vollständig installierten Server, um einen Nano-Server oder um eine Core-Installation handelt. Auf einem herkömmlichen Server verwenden

Sie dazu den Server-Manager oder die PowerShell. Auf einem Core-Server installieren Sie das Feature vor allem in der PowerShell. Dazu verwenden Sie den Befehl *Install-WindowsFeature Containers*. Beim Erstellen eines neuen Nano-Servers lassen sich weitere Optionen in das Image einbinden. Dadurch lassen sich auch die Container-Funktionen installieren. Dazu verwenden Sie die Option *-Containers*. Mehr zu diesem Thema lesen Sie in Kapitel 2.

Anschließend benötigen Sie ein Image, auf dessen Basis Container erstellt werden können. Hier kommt entweder eine Core-Installation oder ein Nano-Server-Image zum Einsatz. Die Verwaltung erfolgt normalerweise mit der PowerShell, alternativ mit dem Docker-Client, den Sie über die PowerShell herunterladen können.

Mit dem Windows-Docker-Client können Sie Container verwalten. Der Docker-Client dient nur zur Verwaltung der Container-Technologie, die direkt in Windows Server 2016 integriert ist. Er stellt selbst keinen Serverdienst zur Verfügung. Der Client kann die Windows Server-Container verwalten, zusätzlich aber auch andere Hosts, zum Beispiel Linux-Server.

In Windows Server 2016 ist der Docker-Client ebenfalls integriert und steht über die Eingabeaufforderung zur Verfügung.

Um Windows Server-Container zu verwalten, installieren Sie die notwendigen Erweiterungen auf dem Server. Dazu muss der Server über eine Internetverbindung verfügen:

Install-Module -Name DockerMsftProvider -Force

Install-Package -Name docker -ProviderName DockerMsftProvider -Force

Restart-Computer -Force

Achtung Achten Sie darauf, dass nach dem Neustart des Servers zusätzlich der Docker-Dienst gestartet werden muss. Rufen Sie dazu das Dienste-Fenster über »services.msc« im Suchfeld des Startmenüs auf und nehmen Sie die entsprechende Einstellung vor.

Anschließend steht der Server bereit und Sie können mit Containern arbeiten. Administratoren, die Docker mit PowerShell DSC installieren wollen, können folgende Befehle verwenden:

Install-Script -Name Install-DockerOnWS2016UsingDSC

Install-DockerOnWS2016UsingDSC.ps1

Tipp Für Entwickler und Administratoren kann es interessant sein, Hyper-V für Container-Hosts auch in Windows 10 zu nutzen. Dazu sind auf dem Windows 10-Host einige Befehle notwendig:

Netsh advfirewall firewall add rule name="docker engine" dir=in action=allow protocol=TCP localport=2375

Stop-Service docker

Dockerd --unregister-service

Dockerd -H npipe:// -H 0.0.0.0:2375 --register-service

Start-Service docker

Erste Schritte mit Docker in Windows Server 2016

Der Befehl *Docker images* zeigt zum Beispiel die vorhandenen Docker-Images auf dem Windows-Server an. Standardmäßig sind noch keine Images vorhanden.

Tipp Erhalten Sie bei der Ausführung des Docker-Befehls eine Fehlermeldung, dass die Authentifizierung fehlt, müssen Sie sich zuerst mit *Docker login* mit Ihrer Docker-ID anmelden. Eine Docker-ID erhalten Sie auf der Internetseite von Docker (*http://tinyurl.com/jrjfsoj*).

Um ein Image auf Basis von Windows Server 2016 zur Verfügung zu stellen, können Sie die notwendigen Daten direkt bei Microsoft/Docker herunterladen:

Docker pull microsoft/windowsservercore

Alternativ stehen folgende Befehle zur Verfügung:

Docker pull microsoft/windowsservercore:10.0.14393.321

Docker tag microsoft/windowsservercore:10.0.14393.321 microsoft/windowsservercore

Sie können als Image für Docker-Container in Windows Server 2016, neben der Core-Installation, auch eine Nano-Installation verwenden. In diesem Fall geben Sie den folgenden Befehl ein:

Docker pull microsoft/nanoserver

Wollen Sie einen Container erstellen und starten, verwenden Sie den Befehl *Docker run*. Der Befehl startet das festgelegte Image als Container. So können Sie sicherstellen, dass das Image funktioniert.

Docker run microsoft/windowsservercore

Tipp Mit dem Docker-Client durchsuchen Sie den Docker-Hub nach Images auf Basis von Windows Server 2016. Dazu verwenden Sie zum Beispiel den Befehl:

Docker search Microsoft

Auch ein Webserver auf Basis der Internetinformationsdienste (IIS) in Windows Server 2016 lässt sich als Container bereitstellen:

Docker run -it -p 80:80 microsoft/iis cmd

Nach dem Start steht der Webserver mit der Standardwebseite bereit. Um die Standardwebseite im Container zu löschen, verwenden Sie zum Beispiel:

Del C:\inetpub\wwwroot\iisstart.htm

Wollen Sie die Startseite mit einer eigenen Seite ersetzen, verwenden Sie den folgenden Befehl:

Echo "Test für IIS im Windows Server-Container" > C:\inetpub\wwwroot\index.html

Sobald der Container erstellt wurde, können Sie ihn über die Eingabeaufforderung und die PowerShell verwalten. Um sich eine Liste aller Container auf einem Container-Host anzuzeigen, verwenden Sie den Befehl

Docker ps -a

Tipp Die installierte Docker-Version und den Docker-Client lassen Sie sich mit *Docker version* anzeigen.

Verschiedene Images für Core und Nano nutzen

Welche Images Sie auf einem Container-Host nutzen können, hängt davon ab, auf welcher Installationsvariante von Windows Server 2016 Sie den Container-Host betreiben.

Auf Servern mit grafischer Oberfläche und Core-Servern können Sie das Core-Image und das Nano-Image von Windows Server 2016 verwenden. Auf Nano-Servern steht bei Windows-Servern nur das Nano-Image zur Verfügung, als Hyper-V-Container können Sie aber auch die Core-Server-Variante verwenden.

Windows Server-Container und der Host teilen sich einen einzelnen Kernel, da die Container den Kernel des Container-Hosts nutzen. Dabei muss das Basisimage des Containers mit dem des Hosts übereinstimmen. Windows Server 2016 kennt hier vier Versionierungsgrade: Hauptversion, Nebenversion, Build und die Revision, zum Beispiel »10.0.14393.0«. Die Revisionsnummer wird aktualisiert, wenn Windows-Updates installiert werden. Das Starten von Windows Server-Containern wird verhindert, wenn die Buildnummer nicht übereinstimmt. Das kann zum Beispiel passieren, wenn Sie Vorabversionen von Windows Server 2016 einsetzen oder aktualisierte Container nutzen. Wenn die Buildnummer übereinstimmt, die Revisionsnummer aber unterschiedlich ist, wird der Container zwar gestartet, allerdings ist der produktive Betrieb nicht empfohlen und wird von Microsoft auch nicht unterstützt.

Sie können in der Registry im Pfad *HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion* erkennen, welche Version auf einem Container-Host installiert ist. Stellen Sie sicher, dass die Tags auf *Docker Hub* oder die Image-Hash-Tabelle in der Beschreibung des Image zu der Version des Hosts passt.

Hinweis Hyper-V-Container verwenden eine eigene Instanz des Windows-Kernels. Daher müssen die Versionen von Container-Host und Container-Image nicht übereinstimmen.

Hyper-V-Container-Host anpassen

Wollen Sie Hyper-V-Container nutzen, benötigen Sie auf dem Container-Host natürlich noch Hyper-V als Serverrolle. Hier haben Sie auch die Möglichkeit, mit einer VM zu arbeiten. In diesem Fall müssen Sie aber für die VM die eingebettete (nested) Virtualisierung konfigurieren. Damit das funktioniert, müssen Sie auf dem Hyper-V-Host, auf dem Sie den Container/Hyper-V-Host betreiben, einige Anpassungen in der PowerShell vornehmen (siehe auch Kapitel 7):

```
#replace with the virtual machine name
$vm = "<virtual-machine>"
#configure virtual processor
```

```
Set-VMProcessor -VMName $vm -ExposeVirtualizationExtensions $true -Count 2
#disable dynamic memory
Set-VMMemory $vm -DynamicMemoryEnabled $false
#enable mac spoofing
Get-VMNetworkAdapter -VMName $vm | Set-VMNetworkAdapter -MacAddressSpoofing On
```

In der virtuellen Maschine, die Sie als Container-Host für Docker und Hyper-V-Container nutzen wollen, können Sie dann noch Hyper-V über die PowerShell installieren:

```
Install-WindowsFeature hyper-v
```

Nano-Server als Container-Host verwenden

Windows Server-Container auf Basis von Docker, aber auch die Hyper-V-Container lassen sich auch auf Nano-Servern betreiben. Sie haben hier zusätzlich die Möglichkeit, den Nano-Server zu virtualisieren. Wollen Sie auf dem Nano-Server auch Hyper-V-Container nutzen, müssen Sie in diesem Fall die eingebettete Virtualisierung aktivieren, so wie im vorangegangenen Abschnitt erläutert. In den Kapiteln 2 bis 4 und 7 sind wir bereits auf diese Themen eingegangen. Generell kann es sinnvoll sein, den Nano-Server in die Domäne mit aufzunehmen.

Eine Remote-PowerShell-Sitzung mit dem Nano-Server erstellen

Um die Container auf einem Nano-Server zu verwalten, sollten Sie am besten über eine RDP-Sitzung des Hyper-V-Hosts, auf dem Sie den Nano-Server virtualisieren, eine Remote-PowerShell-Sitzung zum Nano-Server aufbauen.

Fügen Sie im ersten Schritt den Nano-Server zu den vertrauenswürdigen Hosts auf dem Hyper-V-Host hinzu. Dazu verwenden Sie die IP-Adresse des Nano-Servers. Diese erfahren Sie auch über die Nano Server Recovery Konsole (siehe die Kapitel 2 und 3).

Hinweis In diesem und den folgenden Beispielen hat der Nano-Server die IP-Adresse *192.168.178.225*, der Server trägt die Bezeichnung *nano-hyperv* und ist Mitglied der Domäne *joos.int*.

```
Set-Item WSMan:\localhost\Client\TrustedHosts 192.168.178.225 -Force
```

Danach erstellen Sie die Remote-PowerShell-Sitzung:

```
Enter-PSSession -ComputerName 192.168.178.225 -Credential joos\Administrator
```

Alle Befehle, die Sie jetzt eingeben, werden auf dem Nano-Server ausgeführt.

Windows-Updates auf Nano-Servern installieren

Damit Container auf Nano-Servern funktionieren, müssen Sie die neuesten Updates für Windows Server 2016 installieren. Das gilt natürlich genauso für Core-Server und Server mit grafischer Benutzeroberfläche. Auf Nano-Servern ist die Installation von Windows-Updates teilweise etwas komplizierter. Geben Sie zur Installation von Windows-Updates die folgenden Befehle in einer Remote-PowerShell-Sitzung ein:

$sess = New-CimInstance -Namespace root/Microsoft/Windows/WindowsUpdate -ClassName MSFT_WUOperationsSession

Invoke-CimMethod -InputObject $sess -MethodName ApplyApplicableUpdates

Nachdem alle Updates installiert sind, können Sie den Nano-Server über die Remote-PowerShell-Sitzung neu starten:

Restart-Computer -Force

Docker auf Nano-Servern installieren

Auch auf Nano-Servern müssen Sie Docker installieren, um Windows Server-Container zu nutzen. Verbinden Sie sich dazu nach der Installation der neuesten Updates mit einer Remote-PowerShell-Sitzung und installieren Sie Docker:

Install-Module -Name DockerMsftProvider -Repository PSGallery -Force

Install-Package -Name docker -ProviderName DockerMsftProvider

Nachdem die Installation abgeschlossen ist, starten Sie den Nano-Server neu:

Restart-Computer -Force

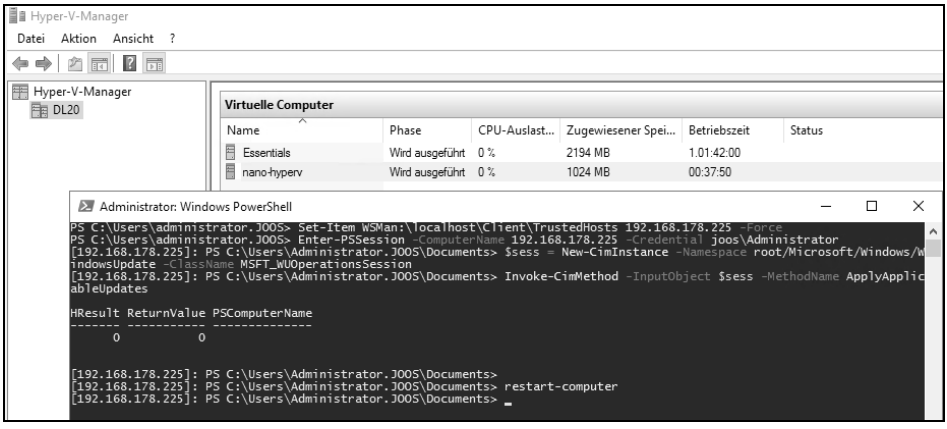

Abbildung 26.1: Installieren von Windows-Updates auf Nano-Servern über eine Remote-PowerShell-Sitzung

Basis-Container-Images auf dem Nano-Server integrieren

Um ein Container-Image auf Nano-Servern zu installieren, verwenden Sie den folgenden Befehl:

Docker pull microsoft/nanoserver

Wollen Sie auch Hyper-V-Container verwenden, müssen Sie Hyper-V auf Ihrem Nano-Server installieren. Wie Sie dazu vorgehen, zeigen wir in den Kapiteln 2 und 7. Anschließend können Sie das Container-Image für Core-Server auf den Nano-Server herunterladen:

Docker pull microsoft/windowsservercore

```
PS C:\Users\administrator.JOOS> Enter-PSSession -ComputerName 192.168.178.225 -Credential joos\Administrator
[192.168.178.225]: PS C:\Users\Administrator.JOOS\Documents> Install-Module -Name DockerMsftProvider -Repository PSGalle
ry -Force

NuGet provider is required to continue
PowerShellGet requires NuGet provider version '2.8.5.201' or newer to interact with NuGet-based repositories. The NuGet
 provider must be available in 'C:\Program Files\PackageManagement\ProviderAssemblies' or
'C:\Users\Administrator.JOOS\AppData\Local\PackageManagement\ProviderAssemblies'. You can also install the NuGet
provider by running 'Install-PackageProvider -Name NuGet -MinimumVersion 2.8.5.201 -Force'. Do you want PowerShellGet
to install and import the NuGet provider now?
[Y] Yes  [N] No  [?] Hilfe (Standard ist "Y"): y
[192.168.178.225]: PS C:\Users\Administrator.JOOS\Documents> Install-Package -Name docker -ProviderName DockerMsftProvid
er

The package(s) come(s) from a package source that is not marked as trusted.
Are you sure you want to install software from 'DockerDefault'?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [?] Hilfe (Standard ist "N"): y
WARNUNG: KB3176936 or later is required for docker to work. Please ensure this is installed.
WARNUNG: Restart is needed to complete installation
WARNUNG: A restart is required to start docker service. Please restart your machine.
WARNUNG: After the restart please start the docker service.

Name                           Version          Source           Summary
----                           -------          ------           -------
Docker                         1.12.2-cs2-ws... DockerDefault    Contains the CS Docker Engine for use with Windows ...


[192.168.178.225]: PS C:\Users\Administrator.JOOS\Documents>
```

Abbildung 26.2: Docker installieren Sie über die PowerShell des Hyper-V-Hosts auf dem virtuellen Nano-Server.

Besonderheiten beim Betrieb von Docker unter Nano-Server

Für den Betrieb von Docker auf einem Nano-Server sind einige Besonderheiten zu berücksichtigen.

Erstellen Sie auf dem Nano-Server eine Firewallregel für die Docker-Verbindung. Bei unsicheren Verbindungen wird Port 2375 verwendet, bei sicheren Verbindungen Port 2376. Auch diese Befehle geben Sie wieder in der Remote-PowerShell-Sitzung ein:

Netsh advfirewall firewall add rule name="Docker daemon " dir=in action=allow protocol=TCP localport=2375

Erstellen Sie eine *daemon.json*-Datei auf dem Nano-Server-Host:

New-Item -Type File c:\ProgramData\docker\config\daemon.json

Geben Sie den folgenden Befehl ein, um die entsprechenden Daten in die Datei einzutragen:

Add-Content 'c:\programdata\docker\config\daemon.json' '{"hosts": ["tcp://0.0.0.0:2375", "npipe://"] }'

Starten Sie den Docker-Dienst neu:

Restart-Service docker

Einen Docker-Client installieren

Um auf dem Hyper-V-Host, auf dem Sie den Nano-Server installiert haben, die Container zu verwalten, benötigen Sie den Docker-Client auch auf dem Hyper-V-Host. Dazu geben Sie folgende Befehle in der PowerShell ein:

Invoke-WebRequest "https://download.docker.com/components/engine/windows-server/cs-1.12/docker.zip" -OutFile "$env:TEMP\docker.zip" -UseBasicParsing

Expand-Archive -Path "$env:TEMP\docker.zip" -DestinationPath $env:ProgramFiles

$env:path += ";c:\program files\docker"

[Environment]::SetEnvironmentVariable("Path", $env:Path + ";C:\Program Files\Docker", [EnvironmentVariableTarget]::Machine)

Anschließend können Sie über den Hyper-V-Host und den Docker-Client auf die Docker-Installation des Nano-Servers zugreifen:

Docker -H tcp://<IP-Adresse des Nano-Servers>:2375 run -it microsoft/nanoserver cmd

---

Tipp Mit dem Befehl *Install-Module posh-docker* laden Sie die automatische Vervollständigung für den Docker-Client auf einen Rechner. Nach der Installation können Sie mit der ⇆-Taste durch die Befehle und Optionen des Docker-Clients schalten. In der PowerShell importieren Sie das Modul mit *Import-Module posh-docker*.

Hyper-V-Container auf Nano-Servern nutzen

Auf Nano-Servern können Sie Container mit dem Container-Image auf Basis des Nano-Image erstellen. Sie können dazu nicht das Core-Image verwenden. Setzen Sie aber auf Hyper-V-Container, können Sie auch das Core-Image als Vorlage für Windows Server-Container verwenden. Dazu müssen Sie jedoch auf dem Nano-Server zuvor Hyper-V installiert haben. Den Befehl können Sie ebenfalls in einer Remote-PowerShell-Sitzung durchführen:

Install-NanoServerPackage Microsoft-NanoServer-Compute-Package

Auch hier müssen Sie den Nano-Server danach neu starten:

Restart-Computer -Force

Erweiterte Konfiguration von Containern durchführen

Sobald Sie den Container-Host installiert und gestartet haben, können Sie mit Docker bereits Container-Images bei Microsoft herunterladen und starten. Auf Basis von Containern lassen sich schnell und einfach eigene Images erstellen. Sie können auch in den Containern Serveranwendungen installieren und bereitstellen.

Container erstellen und Serverdienste verwalten

Docker ermöglicht auch die lokale Verwaltung der Serverrollen. Erstellen Sie zum Beispiel mit dem folgenden Befehl einen neuen Container und wechseln durch Hinzufügen der Option *-it* direkt in die Eingabeaufforderung, können Sie innerhalb des Containers die PowerShell starten und Installationen vornehmen:

Docker run -it --name winiis -p 80:80 microsoft/windowsservercore

Sobald sich die Eingabeaufforderung des Containers öffnet, können Sie mit dem Befehl *Powershell* auf dem Container eine lokale PowerShell-Sitzung öffnen.

Anschließend prüfen Sie zunächst, ob die Internetinformationsdienste (IIS) auf dem Container installiert sind. Dazu verwenden Sie den gleichen Befehl wie bei herkömmlichen Servern mit Windows Server 2016:

Get-WindowsFeature web-server

Um IIS zu installieren, verwenden Sie wiederum den folgenden Befehl:

Install-WindowsFeature web-server

Sobald IIS im Container installiert ist, können Sie über den Befehl *Ipconfig* die IP-Adresse des Containers anzeigen lassen und zum Beispiel vom Container-Host aus mit dem Internet Explorer auf die IP-Adresse des Containers zugreifen. Da IIS installiert ist und Sie den Port 80 auf dem Container aktiviert haben, wird die IIS-Startseite angezeigt.

Tipp Mit *Docker inspect <ID>* können Sie erweiterte Informationen für Container sowie die IP-Adresse des Containers abrufen.

Container und eigene Images erstellen

Auch eigene Images können erstellt und bearbeitet werden. Dies erfolgt zum Beispiel auf Basis bestehender Container, die Sie wiederum mit *Docker ps -a* anzeigen lassen:

Docker commit <ID> <Ordner>/meincontainerimage

Beispiel:

Docker commit 662f25d6d835 windowsiis/joosimageiis

In Docker können Sie also auch Container mit bereits installierten Anwendungen als neues Image speichern und dieses Image für neue Container verwenden. Ob das Image erstellt wurde, können Sie mit *Docker images* anzeigen lassen.

Um Container zu löschen, verwenden Sie den Befehl *Docker rm <Name des Containers>*, der Befehl *Docker rmi <Name des Image>* löscht Docker-Images.

Um zum Beispiel einen Container mit IIS zur Verfügung zu stellen und auf dessen Basis weitere Images anzulegen, müssen Sie zunächst einen neuen Container erstellen, der auf dem vorgefertigten Image basiert:

Docker run -d -p 80:80 microsoft/iis ping -t localhost

Über den Befehl können Sie auch direkt die Ports aktivieren *(-p)* und sicherstellen, dass die Internetinformationsdienste als Dienst gestartet werden (*-d*). Alle gestarteten Container sehen Sie mit *Docker ps*. Nehmen Sie Änderungen an einem Container vor, können Sie

diesen Container zum Beispiel als neues Image speichern und auf Basis dieses Image weitere Container. Dazu verwenden Sie den Befehl *Docker ps -a*, um sich den Namen des Containers anzuzeigen. Anschließend erstellten Sie das Image mit dem folgenden Befehl:

Docker commit <ID> <Neuer Name>

Dockerfiles für eigene Images erstellen

Auf Basis dieses Image erstellen Sie jederzeit weitere Container. Der Vorgang lässt sich automatisieren, indem Sie ein sogenanntes »Dockerfile« verwenden. Dabei handelt es sich um eine Anweisungsdatei für neue Container.

Erstellen Sie dazu ein Verzeichnis auf dem Host und legen Sie darin eine Datei *Dockerfile* (ohne Dateiendung) an. Sie können den Vorgang zum Beispiel mit der PowerShell durchführen:

Powershell New-Item c:\build\Dockerfile -Force

Die Automatisierung nehmen Sie über Befehle in der Datei vor. Dazu müssen Sie die Datei *Dockerfile* in Notepad öffnen:

Notepad c:\build\Dockerfile

In der Datei können Sie zum Beispiel festlegen, dass ein neues Image erstellt werden soll, das IIS als Basis nutzt. In der Datei können Sie auch bestimmen, dass Änderungen an der Konfiguration vorgenommen werden:

FROM microsoft/iis

RUN echo "Dockerfile-Test für automatische Bereitstellung" > c:\inetpub\wwwroot\index.html

Generell können Sie bei Dockerfiles mit der Anweisung *FROM* festlegen, auf welcher Basis der neue Container erstellt werden soll, zum Beispiel mit:

FROM windowsservercore

Mit *RUN* legen Sie fest, was im neuen Container-Image vorgenommen werden soll. Sie können zum Beispiel mit dem folgenden Befehl die Internetinformationsdienste (IIS) in einem neuen Container-Image installieren:

RUN dism.exe /online /enable-feature /all /featurename:iis-webserver /NoRestart

Wollen Sie das Visual Studio-Paket in einem Container installieren, verwenden Sie diesen Aufruf:

RUN start-Process c:\vcredist_x86.exe -ArgumentList '/quiet' -Wait

---

Tipp Sie können über ein Dockerfile auch PowerShell-Skripts in ein Container-Image kopieren und ausführen, zum Beispiel mit:

FROM windowsservercore

ADD script.ps1 /windows/temp/script.ps1

RUN powershell.exe -executionpolicy bypass c:\windows\temp\script.ps1

Um auf Basis dieser Änderungen wiederum ein Image zu erstellen, verwenden Sie in diesem Beispiel diesen Befehlsaufruf:

Docker build -t iis-dockerfile c:\Build

```
PS C:\Users\administrator.JOOS> new-item c:\build\Dockerfile -Force


    Verzeichnis: C:\build


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----       10.11.2016     08:40              0 Dockerfile


PS C:\Users\administrator.JOOS> notepad c:\build\Dockerfile
PS C:\Users\administrator.JOOS> docker build -t iis-dockerfile c:\Build
Sending build context to Docker daemon 2.048 kB
Step 1/2 : FROM microsoft/iis
 ---> 211fecef1e6b
Step 2/2 : RUN echo "Dockerfile-Test f□r automatische Bereitstellung" > c:\inetpub\wwwroot\index.html
 ---> Running in 48574842616d
 ---> a0ad87181b44
Removing intermediate container 48574842616d
Successfully built a0ad87181b44
PS C:\Users\administrator.JOOS>
PS C:\Users\administrator.JOOS> docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
       NAMES
8606f89e8d5b        microsoft/iis       "C:\\ServiceMonitor..."   15 hours ago        Up 15 hours
/tcp   desperate_swartz
PS C:\Users\administrator.JOOS> docker images
REPOSITORY                    TAG                 IMAGE ID            CREATED             SIZE
iis-dockerfile                latest              a0ad87181b44        33 seconds ago      9.51 GB
microsoft/iis                 latest              211fecef1e6b        36 hours ago        9.48 GB
microsoft/iis                 windowsservercore   211fecef1e6b        36 hours ago        9.48 GB
microsoft/sample-dotnet       latest              c9f3d69f9686        2 days ago          908 MB
microsoft/nanoserver          latest              787d9f9f8804        5 days ago          918 MB
microsoft/windowsservercore   latest              f49a4ea104f1        5 days ago          9.2 GB
PS C:\Users\administrator.JOOS>
```

Abbildung 26.3: Docker-Container erstellen und verwalten Sie am besten in der PowerShell.

Sie können erstellte Container mit von Ihnen vorgenommenen Änderungen jederzeit als neues Image speichern und dieses Image wiederum für neue Container verwenden. So erstellen Sie also sehr schnell zahlreiche Container mit allen benötigten Einstellungen. Um ein Image zu erstellen, verwenden Sie zum Beispiel den folgenden Befehl:

Docker commit <ID> meincontainerimage

Sobald Sie das Image erstellt haben, können Sie es mit dem Befehl *Docker images* anzeigen lassen und als Grundlage für einen neuen Container verwenden:

Docker run -it --name dockertest2 meincontainerimage cmd

Container in die Cloud laden (Docker Push)

Mit dem Befehl *Docker pull* laden Sie Container-Images aus Ihrem Docker-Konto auf den Container-Host, um auf Basis des Image einen neuen Container zu erstellen. Sie können aber auch den umgekehrten Weg gehen und Images in Ihr Cloud-Konto hochladen. Der Vorteil dabei ist, dass Sie dieses Image jederzeit wieder herunterladen und auch auf anderen Container-Hosts verwenden können. Sie benötigen dazu eine Docker-ID und müssen sich mit *Docker login* anmelden.

Zum Hochladen von Container-Images verwenden Sie den folgenden Befehl:

Docker push <Benutzername>/iis-dockerfile

Nach dem Upload können Sie mit *Docker pull* das Image auf Container-Hosts herunterladen. Wollen Sie das Image nicht mehr verwenden, können Sie es löschen:

Docker rmi <Benutzername>/iis-dockerfile

Hyper-V-Container in Windows Server 2016 anlegen

Betreiben Sie Docker-Container mit Windows Server 2016 innerhalb von Hyper-V als spezielle Hyper-V-Container, schottet das Betriebssystem diese noch mehr ab als herkömmliche Windows Server-Container auf Basis von Docker. Das erhöht die Sicherheit und Stabilität.

Hyper-V-Container werden – ebenso wie virtuelle Server (siehe Kapitel 7) – über virtuelle Switches an das Netzwerk angebunden. Auch Hyper-V-Container bauen auf Docker auf, bieten aber mehr Möglichkeiten zur Erstellung von Containern.

Der Vorteil der Hyper-V-Container ist eine effizientere Isolierung sowie eine Optimierung der Umgebung für Hyper-V. Hyper-V-Container sind immer von anderen Containern und dem Host isoliert. Da Windows Server-Container Teile des Betriebssystems mit dem Host teilen, besteht das Problem, dass ein Container einen ganzen Host und andere Container beeinträchtigen kann. Mit Hyper-V-Containern ist das nicht möglich, da das Betriebssystem isoliert und virtualisiert wird. Dies ermöglicht es, Container mit Anwendungen auszuführen, die in »Lower Trust«-Umgebungen für Angriffe anfällig sind. Beispielsweise würde dies auf Webserver zutreffen.

Hyper-V-Container verstehen

Windows Server-Container teilen sich wichtige Bereiche des Betriebssystems mit dem Host und anderen Containern. Dadurch erhöht sich zwar im Vergleich zu virtuellen Servern die Effizienz der Container, bietet aber auch mögliche Angriffsflächen. Grundsätzlich ist es möglich, dass ein Container andere Docker-Container auf dem Host beeinträchtigt oder angreift. Der Nachteil von Hyper-V-Containern ist eine etwas schlechtere Leistung im Vergleich zu Windows Server-Container. Der Vorteil liegt in der besseren Isolierung der Container. Sie können auch Freigaben des Hosts in Hyper-V-Containern nutzen, zum Beispiel für die Datenspeicherung oder für Installationsmedien. Die Verwaltung von Hyper-V-Containern kann wie bei herkömmlichen Containern über die PowerShell oder die Eingabeaufforderung erfolgen.

In Hyper-V-Containern ist eine eigene Kopie des Betriebssystems integriert. Der Container läuft in einer Art eingeschränkter virtueller Maschine. Zusammen mit Nano-Servern lassen sich dadurch schnelle und sichere Container zur Verfügung stellen, die alle Vorteile von Windows Server 2016 nutzen. Windows Server-Container, Hyper-V-Container und Nano-Server können gemeinsam und parallel eingesetzt werden.

Sie können in Hyper-V-Containern auch Rechte delegieren, zum Beispiel für mandantengestützte Systeme. Die Hyper-V-Container eines Mandanten können miteinander kommunizieren, während die Container der anderen Mandanten abgeschottet sind. Die Abschottung der Gruppen erfolgt durch Hyper-V in Windows Server 2016. Hyper-V-Container lassen sich per Hyper-V-Replikation auf andere Hyper-V-Hosts replizieren und mit Hyper-V-Clustern hochverfügbar betreiben. Die Übertragung von Hyper-V-Containern auf andere Knoten mit der Livemigration ist ebenfalls möglich.

Hinweis Container-Images müssen nicht angepasst werden, um sie auch als Hyper-V-Container zu nutzen. Images für Container lassen sich für herkömmliche Container, aber auch für Hyper-V-Container nutzen. Sie benötigen also keine verschiedenen Images für die unterschiedlichen Einsatzgebiete.

Bei Bedarf können Sie Windows Server-Container mit wenigen Schritten zu Hyper-V-Container konvertieren. Auch der umgekehrte Weg ist jederzeit möglich. Hyper-V-Container können Sie jederzeit wieder in herkömmliche Container konvertieren. Einstellungen und Daten gehen dabei nicht verloren.

Arbeiten Sie mit einem Nano-Server als Container-Host, können Sie in diesem Hyper-V-Container aktivieren. Nach der Aktivierung verfügt der Container über eine virtuelle Hardware. Diese können Sie in der PowerShell des Servers mit *Get-PnpDevice* anzeigen lassen. In Hyper-V-Containern werden Netzwerkadapter und SCSI-Adapter als Hyper-V-Hardware angezeigt.

Hyper-V-Container erstellen und konfigurieren

Haben Sie Container erstellt, können Sie sie mit der PowerShell und dem Docker-Client verwalten. Hier gibt es zunächst keine Unterschiede zwischen Hyper-V-Containern und Windows Server-Containern. Beim Erstellen eines Hyper-V-Containers mit Docker wird der Parameter *--isolation=hyperv* verwendet.

Wollen Sie einen herkömmlichen Container mit Docker zu einem Hyper-V-Container konvertieren, setzen Sie eine Isolierungsmarkierung. Der Befehl sieht dann zum Beispiel folgendermaßen aus:

Docker run --rm -it --isolation=hyperv nanoserver cmd

Die Vorteile lassen sich an einem Beispiel zeigen. Erstellen Sie mit dem folgenden Befehl einen Container und lassen darin einen dauerhaften Ping-Befehl laufen, ist der Prozess auf dem Host selbst zu erkennen:

Docker run -d Microsoft/windowsservercore ping localhost -t

Der erfolgreich erstellte Container wird mit *Docker ps* angezeigt. Mit *Docker top <Name des Containers>* lassen Sie sich die Prozesse im Container anzeigen. Den Namen sehen Sie mit *Docker ps.*

```
Usage:  docker top CONTAINER [ps OPTIONS]

Display the running processes of a container

C:\Users\administrator.JOOS>docker top fd4658719413
Name                PID                 CPU                 Private Working Set
smss.exe            3820                00:00:00.218        262.1 kB
csrss.exe           5192                00:00:00.281        974.8 kB
wininit.exe         5152                00:00:00.281        852 kB
services.exe        844                 00:00:00.546        2.183 MB
lsass.exe           848                 00:00:00.437        3.486 MB
svchost.exe         5900                00:00:00.250        2.322 MB
svchost.exe         5504                00:00:00.140        1.761 MB
svchost.exe         2512                00:00:00.312        2.261 MB
svchost.exe         1200                00:00:00.375        5.48 MB
svchost.exe         2292                00:00:00.140        3.002 MB
svchost.exe         3792                00:00:01.125        8.815 MB
svchost.exe         2392                00:00:03.234        4.653 MB
svchost.exe         5356                00:00:00.437        3.604 MB
svchost.exe         2212                00:00:00.015        872.4 kB
CExecSvc.exe        1080                00:00:00.031        733.2 kB
PING.EXE            708                 00:00:00.093        598 kB
TrustedInstaller.exe 3696               00:00:00.078        1.278 MB
TiWorker.exe        4644                00:00:02.500        3.322 MB
```

Abbildung 26.4: Lassen Sie sich die im Container ablaufenden Prozesse anzeigen.

In diesem Beispiel sehen Sie anschließend den Ping-Prozess und dessen ID. Mit dem Befehl *Get-Process -Name ping* lassen Sie sich diese Informationen anzeigen. Dadurch ist zu erkennen, dass der Prozess über die gleiche ID wie im Container verfügt.

Alternativ können Sie einen isolierten Hyper-V-Container mit der Option *--isolation* erstellen:

Docker run -d --isolation=hyperv microsoft/nanoserver ping -t localhost

Auch hier lässt sich jetzt auf dem gleichen Weg die ID des Prozesses für den Ping-Befehl abrufen. Dazu verwenden Sie wieder *Docker top*. Suchen Sie erneut nach dem Prozess auf dem Host, ist dieser allerdings nicht zu sehen. Auf dem Host wird in diesem Fall aber der Prozess einer neuen VM sichtbar. Dabei handelt es sich um den virtuellen Computer, der den Hyper-V-Container kapselt und die ausgeführten Prozesse vor dem Hostbetriebssystem schützt.

Docker, Hyper-V-Container und VMs parallel einsetzen

Neben herkömmlichen Windows Server-Containern und Hyper-V-Containern,können Sie dann auf einem Hyper-V-Host (auch auf einem Nano-Server) virtuelle Maschinen erstellen, die wiederum mit den Containern kommunizieren können. Container-Host und Hyper-V-Host schließen sich also nicht aus.

Herkömmliche Installationen von Windows Server 2016 arbeiten mit Containern und Hyper-V-Containern zusammen, genauso wie Core- oder Nano-Installationen von Windows Server 2016. Die Server und Dienste lassen sich in einem gemeinsamen Netzwerk betreiben, auch zusammen mit anderen Betriebssystemen wie Windows Server 2012/2012 R2 oder Linux.

Windows Server-Container in der PowerShell verwalten

Container können Sie recht einfach über die PowerShell verwalten. Das gilt auch für lokale Installationen von Container-Hosts. Mit dem Befehl *Powershell* starten Sie in der Eingabeaufforderung eine neue PowerShell-Sitzung. Alle Befehle für die Verwaltung von Containern lassen Sie sich mit dem Befehl *Get-Command -Module Containers* auflisten.

Nachdem ein Container gestartet ist, können Sie eine PowerShell-Sitzung öffnen und sich mit dem Container verbinden. Dadurch verwalten Sie auch Einstellungen und Serverdienste im Container. Für den Verbindungsaufbau benötigen Sie die ID des Containers. Diese können Sie zum Beispiel mit *Docker ps* herausfinden.

Für den Verbindungsaufbau verwenden Sie den Befehl *Enter-PSSession*. Zusammen mit der Container-ID sowie der Option *RunAsAdministrator* bauen Sie eine Verbindung auf. Der Container hat eine eigene IP-Adresse erhalten, damit er mit dem Netzwerk/Internet kommunizieren kann. Die Syntax des Befehls sieht folgendermaßen aus:

Enter-PSSession -ContainerId <ID> -RunAsAdministrator

Befehle, die Sie hier eingeben, werden im Container durchgeführt. Mit *Exit* verlassen Sie die Sitzung im Container und arbeiten wieder mit dem eigentlichen Container-Host. Bei der Erstellung neuer Container spielen auch die virtuellen Switches auf dem Host eine Rolle. Diese können Sie in der PowerShell mit *Get-VMSwitch* anzeigen. Container verbinden sich über die virtuellen Switches mit dem Netzwerk.

Sie können in Containern Sitzungen unterbrechen und erneut aufbauen. Bei unterbrochenen Sitzungen laufen die Cmdlets in der Sitzung weiter. Dazu nutzen Sie die Cmdlets *Disconnect-PSSession*, *Connect-PSSession* und *Receive-PSSession*.

Wollen Sie von einer lokalen PowerShell-Sitzung über das Netzwerk Programme auf einem Container starten, verwenden Sie den folgenden Befehl:

Invoke-Command -ContainerId <ID> -RunAsAdministrator -ScriptBlock { <Befehl> } -RunAsAdministrator

Ein Beispiel für die Ausführung ist:

Invoke-Command -ContainerId b2f55c8c-28d7-4c0c-ab2b-9ee62c9ae6ea -RunAsAdministrator -ScriptBlock { ipconfig } -RunAsAdministrator

Zusammenfassung

In diesem Kapitel haben Sie erfahren, wie Sie die neuen Container in Windows Server 2016 nutzen sowie Hyper-V-Container einsetzen. Auch die Installation von Container-Hosts war Thema in diesem Kapitel.

Im nächsten Kapitel erläutern wir Ihnen, wie Sie den Webserver IIS in Windows Server 2016 nutzen. Dieser lässt sich auch in Containern betreiben, aber auch auf Nano-Servern, auf Core-Servern und ebenso auf herkömmlichen Servern mit Windows Server 2016.