Hadoop-Cluster und SMACK-Stack bauen

iX KOMPAKT CLOUD-COMPUTING

Herbst 2016

Ein Sonderheft des Magazins für professionelle Informationstechnik

Auf der Heft-DVD

Cloud zum Ausprobieren

- Mirantis OpenStack
- OpenNebula
- ownCloud
- CoreOS

Cloud fürs Unternehmen

ownCloud:
Online-Speicher im Eigenbau

Infrastruktur:

Windows Nano Server

OpenStack leicht gemacht

OpenNebula – Schlanke Alternative

Anwendungen:

Continuous Delivery in der Cloud

Office as a Service

CRM: Salesforce im Griff

Praxis:

Virtual Private Server: Cloud-Hardware zum Mieten

Chromebooks ohne Google benutzen

Recht und Datenschutz:

Verträge prüfen, Risiken ausschalten

HOSTSERVER
Professionelles Hosting

Managed Hosting und IT-Betrieb

zertifiziert nach ISO 27001 IT-Sicherheit und ISO 9001 Qualitätsmanagement

Wir bieten den zertifizierten Betrieb von IT-Projekten und Anwendungen auf Servern, Clustern, in der Cloud – auch in Ihrem RZ – mit individuellem SLA und ITIL-Service-Management.

Nutzen Sie über 15 Jahre Erfahrung für Ihre IT-Projekte.

- Managed Firewall / VPN / Loadbalancer
- DDoS Schutz
- Multi Datacenter Hosting
- Public- und Private-Cloud

Applikationsbetrieb und Managed Hosting am Standort Deutschland vom Server bis zum IT-Projekt mit Fokus auf Verfügbarkeit, Sicherheit und umfassenden Datenschutz.

Anfragen unter: 030 – 47 37 55 50
www.hostserver.de/it

HOSTING
Made in Germany

denic DE-CIX ecix

HOSTSERVER
Berlin ▪ Marburg ▪ Frankfurt am Main

Alles kann in die Cloud. Alles?

Sicher denken viele bei der Cloud zuerst an Speicher im Internet zum Datenaustausch. Den Trend zu dieser Art des Datentransfers begründete die Anwendung Dropbox des gleichnamigen Anbieters. Ein Unternehmen muss bei der Nutzung dieses Dienstes natürlich bedenken, dass die Informationen das eigene Firmennetz verlassen. Obwohl inzwischen etliche Mitbewerber ebenfalls Cloud-Storage anbieten, funktioniert eine Verlagerung ins eigene Rechenzentrum nur in den wenigsten Fällen. Prominentes Beispiel, bei dem dies möglich ist: ownCloud, das in dieser *iX* kompakt ab Seite 78 vorgestellt wird.

Anwendungen aus der Cloud nehmen jedoch auch andere Formen an. Unter dem Stichwort Software as a Service tummelt sich auf dem Markt vieles, das bisher lediglich als lokale Installation verfügbar war – Office-Pakete, Projektmanagement bis hin zur kompletten Umgebung für App-Entwicklung. Vorteile: Pay per Use und Skalierbarkeit, keine eigene Wartung mehr. Die Abrechnung der Dienste erfolgt in der Regel pro Monat und Zahl der Nutzer. Systemadministratoren des Dienstleisters sichern den Betrieb.

Nachteile: Auch wenn der Aspekt Datensicherheit am deutlichsten zutage tritt, gibt es weitere Neuerungen durch die Verlagerung von IT-Diensten in die Cloud. Die Liste der nötigen Prüfungen verlängert sich. Ähnlich wie beim Outsourcing von Dienstleistungen müssen Anbieter speziellen Anforderungen genügen. Sie müssen Zertifizierungen vorweisen, passende Clients für das genutzte Betriebssystem bereitstellen und Supportmitarbeiter beschäftigen, die die gleiche Sprache beherrschen. Auch die technischen Voraussetzungen im eigenen Haus müssen stimmen – Geräte brauchen bestimmte Schnittstellen, insbesondere die Internetverbindung muss ausreichend Bandbreite besitzen.

Betroffen sind zudem Vertragsgestaltung und Risikobewertung. Durch die Weitergabe von persönlichen Daten in die Hände Dritter gelten besondere rechtliche Vorschriften. Wer viele Dienste von unterschiedlichen Anbietern in Anspruch nimmt, muss sich um die Verwaltung von Rechten und Rollen kümmern („Passt nicht immer“ ab Seite 146). In der Regel impliziert Cloud außerdem, dass man sich Ressourcen mit anderen Kunden teilt – das muss bei der Auswahl der Dienste unter den Gesichtspunkten Performanz und Sicherheit Berücksichtigung finden. Zumindest trifft dies für die Public Cloud zu, bei der die Dienste komplett im Rechenzentrum des Anbieters liegen. Mit dem Gegenstück – Private Cloud – holt man sich die IT zurück und gewinnt wieder die volle Kontrolle über seine Daten, muss jedoch auch die Administration übernehmen.

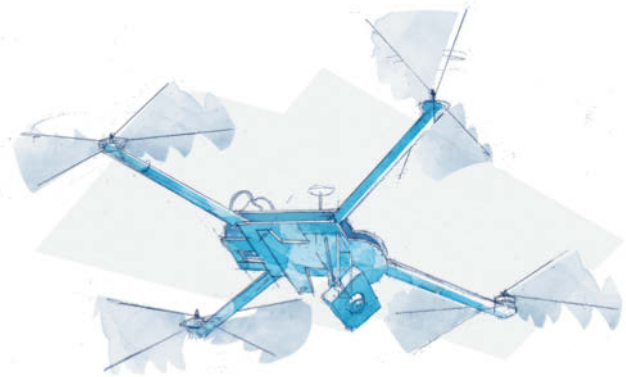
Welche Manifestation der Cloud auch immer man wählt – als Infrastruktur oder nur in Form von Softwarediensten; als Public oder Private Cloud – die Migration dorthin verläuft nicht reibungslos und ohne Anstrengung. Die Risikoanalyse kann sogar ergeben, dass ein Wechsel gar nicht lohnt. Für alle Fälle enthält die *iX* kompakt „Cloud fürs Unternehmen“ Material, um für die verschiedenen Aspekte ein Gespür zu entwickeln sowie Hilfestellungen für die ersten Schritte.

JAN BUNDESMANN

Komplexe Cloud-Anwendung im Griff

Wer Salesforce CRM fürs Verwalten seiner Kundendaten in der Cloud verwenden will, muss die zahlreichen Funktionen kennen, Testdaten einspielen und sich mit Apex-Programmierung befassen.

Seiten 106 bis 125



Infrastruktur überblicken

Eine Cloud zu verwalten – ob lokale, Hybrid oder Public –, ist nicht trivial. Freie Management-Werkzeuge helfen dabei, haben jedoch ihre Eigenheiten.

Seiten 42, 58, 66 und 70

Hintergrundwissen

Administration

Praxis

Anwendungen

Das Überall-Büro

Office-Programme, Projektverwaltung und sogar Softwareentwicklung gibt es aus der Cloud. Sie erleichtern die Arbeit im Team. Ein Blick auf die zahlreichen Angebote und Möglichkeiten.

Seiten 126, 130 und 138

Auf der Heft-DVD

Infrastruktur

Mirantis OpenStack 9.0 als Virtual Appliance: Erstellt mit Mirantis Fuel – alles Nötige fürs direkte Ausprobieren. Die Appliance beinhaltet einen Master-Node und drei Slave-Nodes.

OpenNebula 5.0 Sandbox: Die Virtual Appliance erlaubt das unkomplizierte Ausprobieren von OpenNebula. Virtuelle Maschine starten und unter http://localhost:9869 einloggen.

Cloud-Software

Die beliebte Filesharing-Software **ownCloud in Version 9.1** als Virtual Appliance. Auf der DVD befindet sich die Community-Version der Software.

CoreOS 1122.2.0: ein minimales Linux ohne eigene Paketverwaltung. Software wird in Form von Docker-Containern hinzugefügt.

Oracle Virtual Box 5.1.6: zum Abspielen der Virtual Appliances

Literatur & Know-how

„VMware vRealize Automation Handbook“ von Guido, Jens-Henrik und Constantin Söldner (Community-Version, in englischer Sprache)

Der Artikel **„Datensauger“** ab Seite 116 erläutert das Befüllen des Salesforce CRM mit Testdaten. Diese finden sich auf der DVD.

Hinweis für Käufer der digitalen Versionen

- PDF- und iPad-Version: In der *iX*-App finden Sie einen Button zum Download des DVD-Images.
- PDF-E-Book: Folgen Sie im Browser der unter „Alle Links“ angegebenen URL.

Alle Links: www.ix.de/ix1616004

Artikel mit Verweisen ins Web enthalten am Ende einen Hinweis darauf, dass diese Webadressen auf dem Server der *iX* abrufbar sind. Dazu gibt man den *iX*-Link in der URL-Zeile des Browsers ein. Dann kann man auch die längsten Links bequem mit einem Klick ansteuern. Alternativ steht oben rechts auf der *iX*-Homepage ein Eingabefeld zur Verfügung.

CLOUD BACKUP

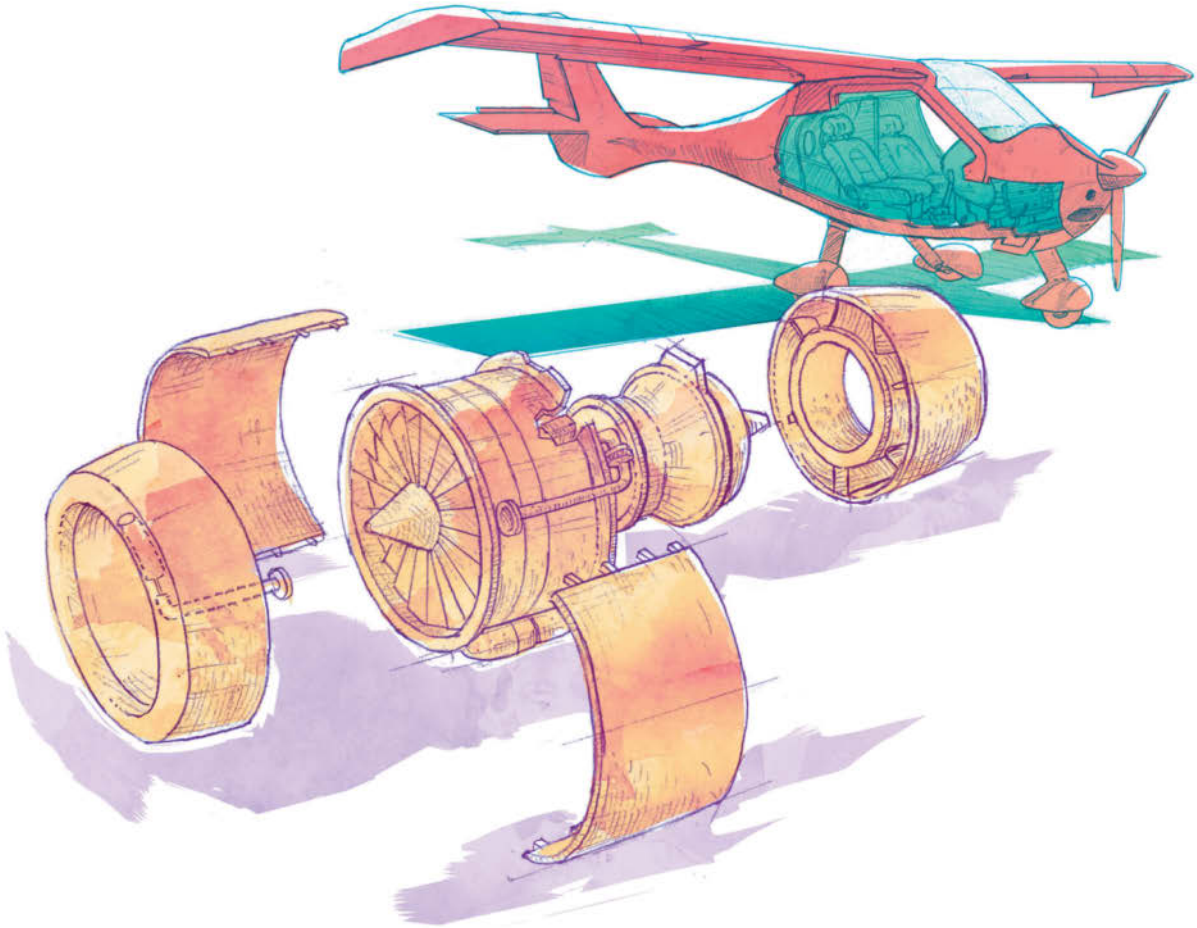
Daten sind Ihr Kapital –
wir sichern dieses regional.

Die transtec Cloud für den Mittelstand bietet Ihnen:

- Datensicherung und Datenverfügbarkeit zu niedrigen Kosten
- Sicherung in ISO-zertifizierte transtec-Rechenzentren
- End-To-End Verschlüsselung der Daten für höchstmöglichen Datenschutz
- Standorte der Rechenzentren ausschließlich in Deutschland mit eigener Hardware

transtec AG
Tel. +49 (0) 71 21 / 26 78 - 400
› sales@transtec.de | www.transtec.de

transtec



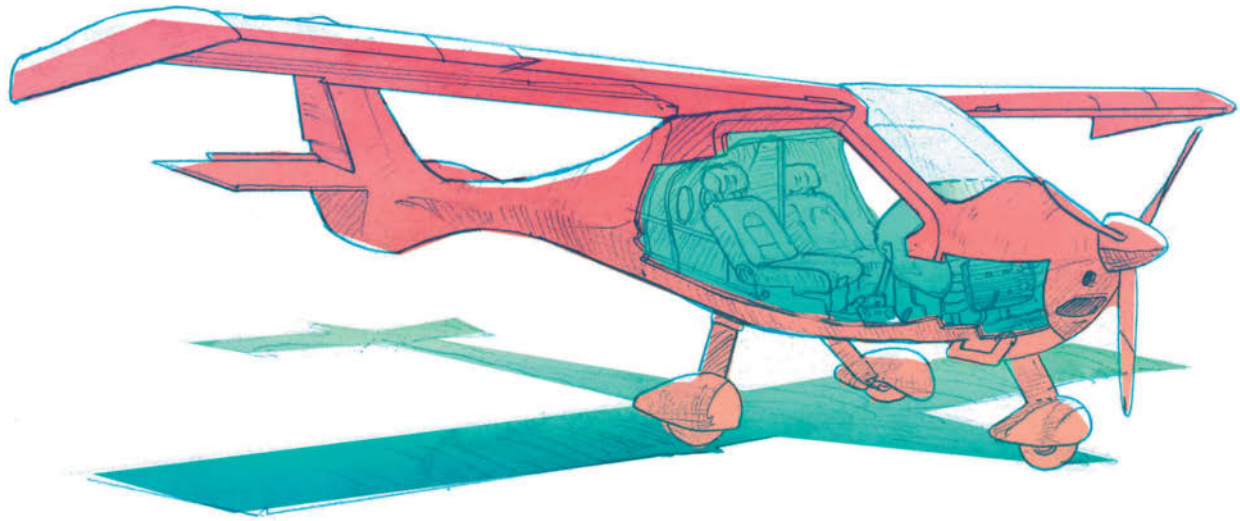
Startvorbereitungen

Mit der Cloud hält neue Technik Einzug ins Rechenzentrum: Betriebssysteme stehen in einer miniaturisierten Variante bereit und betreiben Anwendungen in Containern, zentrale Software steuert die Netzhardware und virtualisiert lokale sowie Weitverkehrsnetze. Auch dem Identitätsmanagement eröffnet die Wolke neue Möglichkeiten.

Microsofts Nano-Variante

Ultraleichtgewicht

Holger Schwichtenberg



Im Rahmen von Windows Server 2016 bietet Microsoft eine „Nano"-Variante, die nur die nötigsten Funktionen enthält. Es gibt weniger laufende Prozesse und offene Ports; ein Neustart dauert daher nur drei Sekunden.

Das Ansinnen von Microsoft, den allmächtigen, aber voluminösen Windows Server zu verschlanken, ist nicht neu. Seit einigen Jahren muss man nach dem eigentlichen Setup benötigte Dienste in Form von Rollen nachinstallieren. Seit 2008 gibt es Windows Server Core, den Microsoft als „Server ohne GUI" anpreist, dabei besitzt dieser weiterhin nicht nur einen Anmeldebildschirm, sondern auch Fenster. Einem Core-Server fehlen jedoch Startmenü, Desktop, Systemsteuerung und alle anderen grafischen Verwaltungs- und Begleitprogramme. Der Unterschied bei Installationszeit, Boot-Zeit, Festplattenbedarf und Update-Häufigkeit ist bei Server Core jedoch nicht wesentlich geringer als bei einem vollständigen Server.

Genau an diesen Punkten greift das Unternehmen nun mit dem neuen Nano Server an. Dieser dampft die Windows-Funktionen signifikant ein: keine grafische Benutzeroberfläche mehr, sondern lokal nur noch eine textbasierte „Recovery Console" mit reduziertem Funktionsumfang (Abbildung 1).

Bei der ersten Anmeldung kann man darin das Kennwort des Administrators setzen. Danach gibt es nur noch einen eng umgrenzen Satz von Einstellungen. Eine interaktive Eingabe von beliebigen Befehlen ist nicht vorgesehen. Die Steuerung erfolgt komplett über die Tastatur; die Maus ist nicht nutzbar.

Die verfügbaren Einstellungen drehen sich um die Netzwerkkarten und die Firewall sowie das Windows Remote Management (WinRM). Alle Einstellungen in der Recovery Console haben den Fokus auf die Herstellung einer Fernverbindung mit dem Nano Server per PowerShell Remoting, Windows Management Instrumentation (WMI), Microsoft Management Console (MMC) und anderer (RPC-basierter) Windows-Werkzeuge wie dem Server Manager und dem Registry Editor. Eine Verwaltung per Remote Desktop Protocol (RDP) ist ebenso wenig möglich wie das Nutzen einzelner MMC-Funktionen, etwa des Astes „Task Scheduler" in der Computerverwaltung. Im Server Manager steht stets „Windows PowerShell not installed", obwohl diese ja Kernbestandteil von Nano Server ist.

Von allem weniger

Die Reduktion sieht man auch bei den Systemdiensten: Auf Nano Server laufen nur noch 22 Dienste statt 46 wie auf einem Server Core oder 79 auf einem vollständigen Windows Server. Der Neustart eines Nano Servers in einer virtuellen Maschine dauert nur drei Sekunden. Microsoft sieht die VHD-Größe (Virtual

Listing 1: Nano Server als Docker-Image unter Windows Server 2016 herunterladen

```
Install-WindowsFeature containers
Restart-Computer -Force
New-Item -Type Directory -Path 'C:\docker\'
Invoke-WebRequest https://aka.ms/tp5/b/dockerd -OutFile $env:ProgramFiles\docker\dockerd.exe -UseBasicParsing
Invoke-WebRequest https://aka.ms/tp5/b/docker -OutFile $env:ProgramFiles\docker\docker.exe -UseBasicParsing

[Environment]::SetEnvironmentVariable("Path", $env:Path + ";C:\Docker", [EnvironmentVariableTarget]::Machine)
dockerd --register-service
Start-Service docker

Install-PackageProvider ContainerImage -Force
Install-ContainerImage -Name NanoServer
Restart-Service docker
docker images
```

Listing 2: Nano Server als Docker-Image unter Windows 10 herunterladen

```
Enable-WindowsOptionalFeature -Online -FeatureName containers -All
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All
Restart-Computer -Force
New-Item -Type Directory -Path 'C:\docker\'
Invoke-WebRequest https://aka.ms/tp5/b/dockerd -OutFile $env:ProgramFiles\docker\dockerd.exe -UseBasicParsing
Invoke-WebRequest https://aka.ms/tp5/b/docker -OutFile $env:ProgramFiles\docker\docker.exe -UseBasicParsing

[Environment]::SetEnvironmentVariable("Path", $env:Path + ";C:\Docker", [EnvironmentVariableTarget]::Machine)
dockerd --register-service
Start-Service docker

Install-PackageProvider ContainerImage -Force
Install-ContainerImage -Name NanoServer
Restart-Service docker
docker images
```

Hard Disk) um 93 % reduziert, die Anzahl der Hotfixes um 92 % und die der notwendigen Reboots um 80 %. Die Anzahl der geöffneten Ports und damit die Angriffsfläche ist geringer.

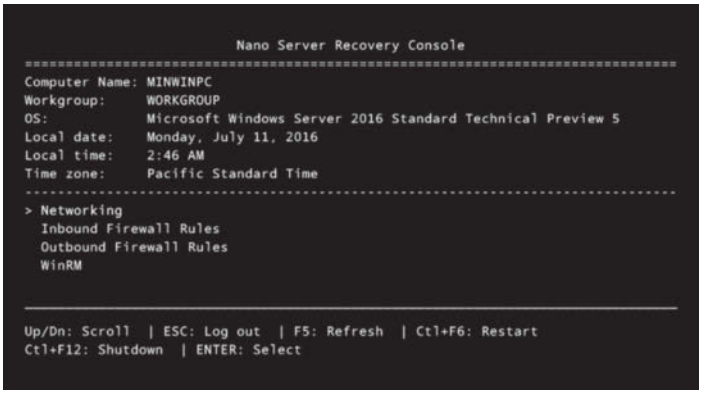
Administratoren installieren den Nano Server nicht mehr über ein GUI, sondern müssen die Konsole bemühen. Die ISO-Datei des Windows Server 2016 stellt im Ordner */NanoServer/NanoServerImageGenerator* ein PowerShell-Modul bereit. Mit dem Commandlet *New-NanoServerImage* erstellt er eine VHD-, VHDX- oder WMI- Datei, mit der sich ein Nano Server mit den gewünschten Features installieren lässt.

Nur wenige Rollen

Bislang bietet Nano Server nur eine kleine Auswahl von Windows-Rollen an:

- Hosting von Webanwendungen und Webservices in den Internet Information Services (IIS)
- Hosting von virtuellen Maschinen in Hyper-V
- Hosting von Windows-Containern mit Docker
- SMB File Server
- Domain Name Server (DNS)

Mit dem Commandlet *New-NanoServerImage* integrieren Administratoren diese Dienste bereits in die Grundinstallation. Im klassischen Windows Server fügen sie diese erst nach der Installation per Server Manager hinzu. Zudem können sie komplementäre Funktionen aktivieren, etwa Desired State Configuration (DSC) zur deklarativen Serverkonfiguration [1], Failover Clustering und Windows Defender für die Anti-Malware-Prüfung. Auch Hardwaretreiber für Netzwerkkarten und Storage sowie statische IP-Adressen und einen Domänenbeitritt kann man vorab festlegen. Zusätzlich lassen sich beliebige Dateien (etwa Webanwendungsdateien) injizieren und ein Startskript für den ersten Bootvorgang mitgeben.



Textbasierte Recovery Console im Windows Nano Server 2016 (Abb. 1).

Den Nano Server gibt es wie die größeren Varianten in unterschiedlichen Funktionsumfängen: „Standard“ und „Data Center“. Bei Nano Server beherrscht die Data-Center-Variante zusätzlich „Shielded VM“: virtuelle Systeme, die sich nur auf bestimmten Hyper-V-Hosts betreiben lassen und damit einen zusätzlichen Schutz bei der Zweckentfremdung einer VHD-Datei bieten.

VHD-Datei und Docker-Image zum Download

Alternativ zum Zusammenstellen eines eigenen Images bietet Microsoft ein vorinstalliertes virtuelles System als VHD-Datei zum Download an. Dieses lediglich 585 MByte große Abbild bietet zunächst nur die Grundfunktion eines File Servers. Außerdem sind die Gast-Dienste zum Hosting des Nano Servers in einem Hyper-V-Host vorinstalliert. Standardmäßig ist der Port 5985, den PowerShell Remoting verwendet, für WinRM geöffnet. Andere Ports für Ping oder SMB File Sharing (445, 5445) müssen von Hand in der Firewall geöffnet werden. Der Nano Server hat den Standardnamen MINWINPC und bezieht seine IP-Adresse per DHCP.

Auch ein 810 MByte großes Docker-Image stellt Microsoft für Nano Server bereit. Dieses kann man unter Windows 10 und Windows Server 2016 herunterladen. Listing 1 zeigt den PowerShell-Skriptcode für Windows Server 2016; unter Windows 10 läuft das Skript aus Listing 2 ab Build 14352. Auf dieser Basis kann man anschließend verschiedene Docker-Images für Nano Server herunterladen, zum Beispiel mit Apache, MySQL, node.js, Python, Redis und Ruby.

Für die Fernverwaltung eines Nano Servers per Windows PowerShell bauen Administratoren entweder

Listing 3: Aufbau einer interaktiven Sitzung zu einem System, das nicht Teil der Domäne ist

```
$ErrorActionPreference = "stop"
$pc = "10.31.63.71"
$benutzer = "$pc\administrator" #~\
$kennwort = "x"

"WinRM-Einstellungen ändern, um Aufbau zum Nicht-Domänen-Rechner zu erlauben..."
cd WSMan:\localhost\Client
Set-Item AllowUnencrypted true
set-item trustedhosts "$pc" -force -Concatenate
Restart-Service winrm
"WinRM-Einstellungen geändert!"

# Sitzung konfigurieren
$kennwortSecure = ConvertTo-SecureString -String $kennwort -AsPlainText -Force
$cred = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList $benutzer, $kennwortSecure
$s = New-PSSession  -Credential $cred  -ComputerName $pc

"Sitzung zum PC $pc aufbauen..."
Enter-PSSession $s
```

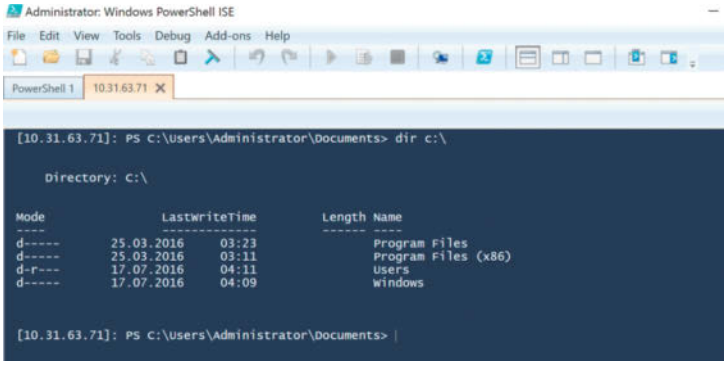
- eine interaktive Sitzung mit dem PowerShell Integrated Scripting Environment (ISE) im Menü „File/New Remote PowerShell Tab“ oder
- eine interaktive Sitzung mit dem Commandlet Enter-PSSession auf oder
- sie senden einzelne Befehle oder ganze Skripte mit Invoke-Command zum entfernten System.

Fernverwaltung mit PowerShell

In allen Fällen ist der Name des entfernten Systems im Parameter *-Computername* und gegebenenfalls ein vom angemeldeten Benutzer abweichendes Benutzerkonto im Parameter *-Credential* anzugeben. Wenn das entfernte System nicht Teil der gleichen Windows-Domäne ist, sind zusätzliche Einstellungen notwendig, die Listing 3 zeigt.

Abbildung 2 zeigt den erfolgreichen Aufbau einer interaktiven Sitzung zum Nano Server im ISE. Die PowerShell zeigt die Fernverbindung durch Voranstellen von Rechnername oder IP-Adresse in eckigen Klammern vor dem Prompt an. Im Fall eines „Remote PowerShell Tab“ sieht man die gleiche Bezeichnung im Titel der Registerkarte. Die im Nano Server verfügbaren PowerShell-Commandlets listet man mit *Get-Command* auf. Viele klassische Windows-Kommandozeilenbefehle wie *netsh* und *ping* sind verfügbar. Aus einer interaktiven Sitzung innerhalb des ISE heraus kann man mit dem Befehl *psedit* eine Datei auf dem entfernten System bearbeiten.

Administratoren können eine Nano-Server-Installation auch im laufenden Betrieb erweitern. Dazu nutzen sie nicht wie bisher den Server Manager oder dessen PowerShell-Commandlet, sondern die in PowerShell 5.0 eingeführte Paketverwaltung



Remote PowerShell Tab zu einem Nano Server im PowerShell ISE (Abb. 2)

OneGet. Dies müssen sie auf einem Nano Server jedoch erst mit der Befehlsfolge

```
Install-PackageProvider NanoServerPackage
Import-PackageProvider NanoServerPackage
```

aktivieren. Hierzu ist ein Verbindung zu www.powershellgallery.com und www.oneget.org erforderlich.

Danach finden Administratoren mit *Find-NanoServerPackage* Installationspakete in Microsofts Online-Repository auf dem Server https://az880830.vo.msecnd.net. Mit *Save-NanoServerPackage* lädt man ein Paket herunter, ohne es zu installieren. Die Installation erfolgt mit *Install-NanoServerPackage*. Für die Installation von Hyper-V auf Nano Server ist das Paket mit dem nicht ganz naheliegenden Namen *Microsoft-NanoServer-Compute-Package* zu verwenden: *Install-NanoServerPackage Microsoft-NanoServer-Compute-Package*. Im Fall der Hyper-V-Installation fordert Nano Server anschließend zum Neustart auf.

Abgespeckter IIS

Der Befehl *Get-Package -ProviderName NanoServerPackage* liefert eine Liste der installierten Pakete. Neben den eigenen Paketen versteht Nano Server weitere Systeme: Windows-Server-App-Pakete, Nuget-Pakete, PowerShellGet-Pakete und Container-Images. Ein Eintrag in Microsofts Techblog verweist auf die verstreuten Dokumentationen zu den einzelnen Paketmanagern [a]. Softwareinstallation per Microsoft Installer (MSI) funktioniert auf Nano Server nicht.

Microsofts Webserver *Internet Informationen Services* (IIS) erfuhr umfangreiche Anpassungen, um auf Nano Server zu laufen. Der *IIS Nano* basiert auf dem IIS 10.0 in Windows 10 und Windows Server 2016 und bietet weder klassische Active Server Pages (ASP) noch ASP.NET Webforms oder ASP.NET MVC auf Basis des .NET „Full“ Framework 1.0 bis 4.6, sondern nur das neue ASP.NET Core. Technology Preview 5 (TP5) enthält eine Vorabversion von .NET Core im Verzeichnis *\\10.31.63.71\c$\Windows\System32\DotNetCore\v1.0*. Die endgültige Version von Nano Server wird auf der am 27.6.2016 erschienenen Version 1.0 aufsetzen.

Für den Betrieb von ASP.NET-Core-Anwendungen werden zusätzliche Komponenten benötigt [b], was aktuell noch mit vielen manuellen Schritten verbunden ist – insbesondere, weil das Installationspaket „.NET Core Windows Server Hosting“ noch nicht auf Nano Server ausgelegt ist. Neben ASP.NET-Core-Webanwendungen lassen sich PHP (ab Version 7), node.js und Python (Django) im IIS Nano installieren. Als Datenbank steht bisher nur MySQL zur Verfügung [c].

bluechip

CLOUD SERVICES

Exklusiv für iX-Leser. Jetzt bluechip Cloud Fachhandelspartner* werden und die bluechip Cloud Services bis Ende Oktober kostenfrei testen.

Einfach kostenfrei registrieren und bei der Bestellung den Voucher-Code "IX-LESER" eingeben.

https://ix.bluechip-cloud.de

- **Datenschutz im höchsten Maße:** deutsches Unternehmen im deutschen Rechenzentrum
- **Kein Risiko:** 1 Monat Mindestvertragslaufzeit für bluechip Cloud Dienste
- **Einfacher Einstieg:** intuitiv zu bedienende Buchungs- und Administrations-Weboberflächen
- **Datensicherheit:** automatische Backups der letzten vier Wochen
- **Werden Sie ihr eigener Cloud Provider:** Ihre eigene Cloud-Plattform so einfach wie noch nie!

Laut aktuellen Studien setzen sich über 85% der deutschen mittelständischen Unternehmen mit dem Thema Cloud intensiv auseinander. Die bluechip Cloud Services eröffnen Ihnen hierbei die Möglichkeit, individuell auf Ihr oder das Unternehmen Ihrer Kunden zugeschnittene Virtual Private-, Public-, Hybrid-, Multi- und Managed Cloud Dienste anzubieten. Die einzigartigen Buchungs- und Administrations-Weboberflächen für Fachhandelspartner, Systemhäuser aber auch für Endanwender sowie die kurze Vertragslaufzeit von nur einem Monat ermöglichen die schnelle und flexible Reaktion auf sich stetig verändernde Marktsituationen. Gepaart mit über 24 Jahren Praxiserfahrung im Bereich Hardware kann das kompetente bluechip Cloud- und Consulting-Team Sie zudem bei Bedarf nach eigenen Private Cloud-Lösungen zu diesem Thema individuell beraten und gemeinsam mit Ihnen die passend auf Sie zugeschnittene Lösung finden.

- **Standort Deutschland:** unmittelbare Nähe zu bluechip in Falkenstein/Vogtland
- **Sicherheit:** hochmoderne Videoüberwachung gepaart mit strengen Zutrittskontrollen
- **Verfügbarkeit:** redundante Stromversorgung, USV-Anlagen und Internet-Anbindungen
- **Umweltbewusstsein:** energieeffiziente und redundante freie Kühlung sowie Verwendung von Ökostrom
- **Brandschutz:** modernes Brandfrüherkennungssystem

bluechip Cloud Services werden in einem namhaften deutschen Rechenzentrum angeboten. Bei der Auswahl des Betreibers wurde ein besonderes Augenmerk auf ein deutsches Unternehmen gelegt, welches bereits nach den gängigen deutschen Datenschutzrichtlinien arbeitet. Dies ist eine Grundvoraussetzung, um eine Zertifizierung nach dem ISO 27001 Standard zu erhalten, die bei bluechip Cloud Services erfüllt wird. Der Zutritt zu der Hardware wird bereits an der Einfahrt zum Gelände kontrolliert und nur mit entsprechender Authentifizierung erlaubt. Ausschließlich ausgewählte bluechip Mitarbeiter erhalten den Zugriff auf die bluechip eigene Hardware. Um die Qualität der bluechip Cloud Services gewährleisten zu können, verfügt die bluechip Hardware im Rechenzentrum sowohl über redundante Stromversorgung samt USV-Anlagen und Dieselgeneratoren sowie über redundante Kühlungs- und Klimatisierungslösungen als auch über eine redundante Anbindung ins Internet über zwei getrennte Router.

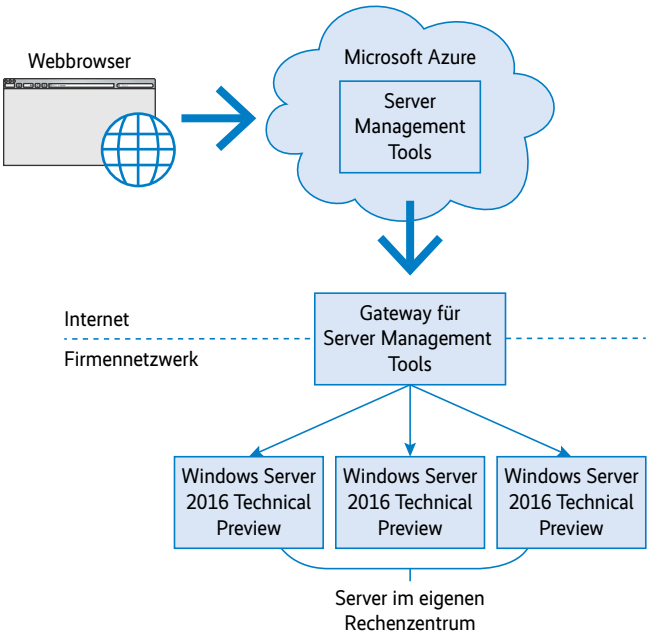
- **Neueste Technologie:** neueste Intel® Xeon™ E5 v4 Prozessoren im Einsatz
- **bluechip SERVERline:** robuste und redundante bluechip Hardware im Cloud-Einsatz
- **bluechip STORAGEline:** redundantes Storage-Cluster mit automatischem Failover
- **Infrastruktur:** redundante High-End-Netzwerke mit 1 Gbit/s-, 10 Gbit/s- und 40 Gbit/s-Verbindungen
- **Datacenter SSDs:** basierend auf modernsten Samsung Datacenter SSDs

Neben dem ausgezeichneten Datenschutz und höchster Datensicherheit ist es wichtig, ständig auf Ihre Dienste in der bluechip Cloud zugreifen zu können. Nur so kann Flexibilität, Agilität und Individualität gewährleistet werden. Deshalb betreibt bluechip die Cloud Services auf eigenen SERVERline und STORAGEline Systemen mit Intel® Xeon™ Prozessoren, die redundant miteinander zu Clustern verbunden wurden. Bei der Architektur des Service-Angebots wurde besonders darauf geachtet, keine Single-Points-of-Failure zuzulassen. Langjährige Partnerschaften der bluechip im Bereich Hardware erlauben es, nicht nur die Verfügbarkeit zu erhöhen, sondern auch die Performance zu steigern. So werden z.B. die Dienste Hosted-Exchange und Infrastructure-as-a-Service mit der SSD Storage Option auf neuesten Samsung Datacenter SSDs betrieben. Alle Festplatten und SSDs sind zudem über aktuellste 12 Gbit/s SAS RAID-Controller mit CacheVault Optionen vor Ausfall und Datenverlust abgesichert.

bluechip Computer AG
Geschwister-Scholl-Str. 11a · 04610 Meuselwitz

Tel.: 03448 755-0
www.bluechip.de

*bluechip vermarktet die Cloud Services ausschließlich über Fachhändler und Systemhäuser aus dem IT-Bereich.



Fernverwaltung von Windows Server 2016 (inkl. Nano Server) mit den Server Management Tools in Azure (Abb. 3).

Bei der Verwaltung des IIS gibt es gravierende Änderungen: Die MMC-basierte IIS Management Console lässt sich nicht einsetzen, da sie mit dem IIS Management Service (wmsvc) kommuniziert – den gibt es in IIS Nano nicht. An dessen Stelle sollen zukünftig eine REST-basierte API und eine webbasierte Oberfläche treten. Derzeit kann man Websites über die PowerShell konfigurieren, jedoch nicht über das bisher verwendete „WebAdministration"-Modul, sondern über das in Windows 10 neu eingeführte Modul „IISAdministration". Man legt dort eine Website etwa mit *New-IISSite* statt mit *New-Website* an.

Ein Nano Server kann keine 32-Bit-Anwendungen, sondern nur 64-Bit-Anwendungen betreiben. Von den Windows-APIs steht deshalb lediglich ein Teil zur Verfügung. Eine Liste findet sich online [d]. Von der .NET-Klassenbibliothek stehen nur die wenigen in .NET Core 1.0 enthaltenen Klassen zur Verfügung. Microsoft stellt mit *NanoServerApiScan.exe* ein Kommandozeilenwerkzeug bereit, das ausführbare Windows-Dateien (DLL, EXE) daraufhin untersucht, ob sie mit dem Nano Server kompatibel sind [e]. Ob .NET-Anwendungen auf .NET Core laufen, zeigt der .NET API Portability Analyzer [f].

Die PowerShell steht ebenfalls in einer abgespeckten Core-Variante zur Verfügung, die auf .NET Core basiert. Es gibt weniger Commandlets und keine PowerShell-Workflows. Die Skriptsprache steht aber komplett zur Verfügung. Beim Abruf der Variablen *$psversiontable* meldet sich die PowerShell auf Nano Server genauso wie bei den anderen Varianten mit der Versionsnummer 5.1.

iX-Wertung

⊕ Kleine Installationsgröße
⊕ Sehr schneller Systemstart

⊖ Sehr beschränkte Dienste und APIs
⊖ Nur verfügbar für Software-Assurance-Kunden
⊖ Webbasierte Verwaltung bisher nur über die Cloud

Zu bedenken ist auch, dass viele Zusatzwerkzeuge von Microsoft und Drittanbietern nicht auf Basis der reduzierten APIs im Nano Server laufen können und diese Werkzeuge neu implementiert werden müssen. Mittlerweile gibt es einen Teil der Sysinternal Tools von Mark Russinovich schon für Nano Server [g].

Serververwaltung aus der Cloud

Ein neuer Weg zur Fernverwaltung eines Nano Servers sind Microsofts Azure-basierte Server Management Tools (SMT): eine Anwendung in Microsofts Cloud, die über einen Gateway-Dienst mit den Servern – On-Premise oder in Azure gehostet – kommuniziert (siehe Abbildung 3). Das Installationspaket für den Gateway-Dienst erhalten Administratoren im Azure-Verwaltungsportal als individuellen Download (GatewayService.MSI). In diesen sind schon alle benötigen Konfigurationseinstellungen zur Kommunikation mit der Cloud verpackt. Da das Gateway die Kommunikation zu Azure per HTTP aufbaut, sind keine Ports für eingehende Kommunikation in der Firewall zu öffnen. Wenn die zu verwaltenden Rechner nicht Teil einer Windows-Domäne sind, muss der Administrator auf dem Gateway-Server die Kommunikation mit den anderen Rechnern durch den PowerShell-Befehl *Set-item WSMAN:\localhost\client\TrustedHosts* * noch erlauben.

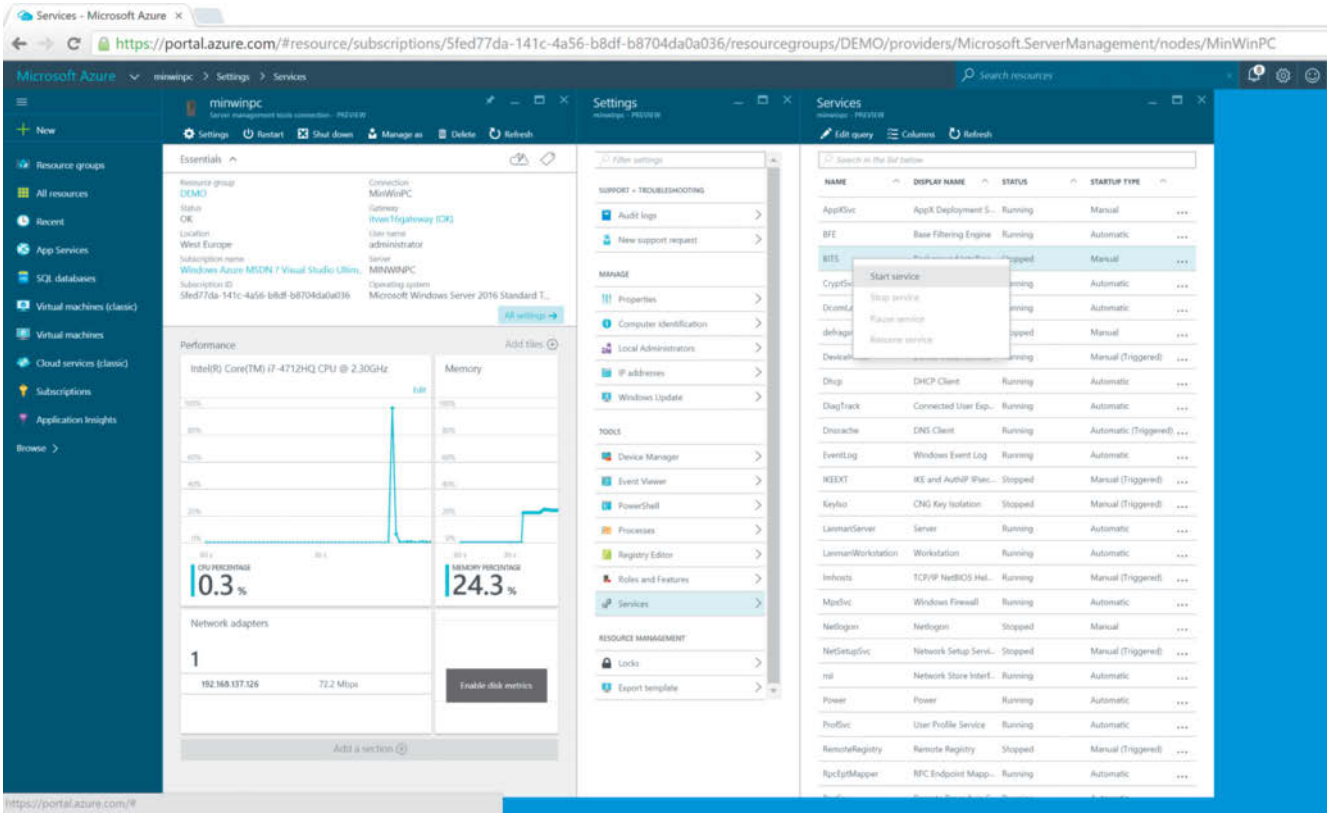
In jedem Fall muss man im Azure-Verwaltungsportal Benutzernamen und Kennwort eines berechtigten Administrators für die einzelnen Server hinterlegen – so viel Vertrauen zu dem US-Konzern ist sicherlich nicht jedermanns Sache. Zusätzlich kann ein Angreifer, der den Zugang zum Azure-Verwaltungsportal knackt, direkt alle dort eingebundenen lokalen Server übernehmen.

Per SMT-Oberfläche kann ein Administrator derzeit noch nicht alle Verwaltungsfunktionen ausführen. Bisher hat Microsoft das Installieren von Rollen und Features sowie Windows Updates, das Auslesen von Hardwareinformation und des Ereignisprotokolls, Einstellungen für Netzwerkkarten, das Starten und Beenden von Prozessen und Systemdiensten (siehe Abbildung 4), die (De-)Aktivierung von Gerätetreibern sowie die Bearbeitung der Registry und die Mitglieder der lokalen Administratorgruppe implementiert.

Für alle anderen Aufgaben bietet SMT eine PowerShell-Konsole im Browser. Hierin steckt die eigentliche Macht der SMT, denn bisher mussten Administratoren für die webbasierte Fernverwaltung PowerShell Web Access (PWA) installieren und dafür eine Schneise in die Firewall schlagen. Microsoft hat noch nicht bekanntgegeben, wann man SMT in seinem lokalen „Azure Stack" – auch ohne Internetzugang und Azure-Konto – nutzen können wird. SMT ist außerdem auf die Verwaltung von Windows Server 2016 beschränkt.

Microsoft nennt den Nano Server in der vorliegenden TP5 „Feature complete" [h]. Die endgültige Version soll im Rahmen der Ignite-Konferenz erscheinen, die am 26.9.2016 beginnt. Während der Redmonder Konzern die großen Server-Varianten wie bisher 10 Jahre mit Updates versorgt, wird es für Nano Server Aktualisierungen in kürzeren Abständen geben. Kunden dürfen maximal zwei Versionen hinterherhinken. Einige der angekündigten, in TP5 noch nicht umgesetzten Funktionen beziehen sich auf Aktualisierungen nach dem Erscheinen. Ein vollautomatisches Aktualisieren von Features erfolgt nicht.

Auch hinsichtlich Windows Server Core hat Microsoft Veränderungen vorgenommen: Der Begriff „Core" kommt im Setup nicht mehr vor. Im Installationsprogramm findet man die Auswahl zwischen „Windows Server 2016 Standard" und „Windows Server 2016 Standard (with Desktop Experience)". Die erste Option ist der bisherige Core-Server, die zweite der klas-



Systemdienste verwalten auf einem lokalen Windows Nano Server 2016 via Server Management Tools in Azure (Abb. 4).

sische Windows Server mit komplettem GUI. Nach der Installation stellt man fest, dass der Core-Server den grafischen Anmeldedialog sowie den STRG+Alt+Entf-Dialog durch eine rein textbasierte Oberfläche ersetzt hat. Wenn man dann aber den Textmanager startet, sieht man, dass der Core-Server auch in Version 2016 noch GUIs darstellen kann.

Onlinequellen

[a] Blogeintrag zu Paketmanagern
https://blogs.technet.microsoft.com/nanoserver/2016/04/27/packagemanagement-support-on-nano-server/

[b] ASP.NET Core on Nano Server
https://docs.asp.net/en/latest/tutorials/nano-server.html

[c] MySQL on Nano Server
https://blogs.technet.microsoft.com/nanoserver/2016/06/13/mysql-on-nano-server/

[d] Nano Server APIs
https://msdn.microsoft.com/en-us/library/mt588480(v=vs.85).aspx

[e] NanoServerApiScan.exe updated for TP5
https://blogs.technet.microsoft.com/nanoserver/2016/04/27/nanoserverapiscan-exe-updated-for-tp5

[f] .NET Portability Analyzer
https://visualstudiogallery.msdn.microsoft.com/1177943e-cfb7-4822-a8a6-e56c7905292b

[g] Sysinternals Suite
https://technet.microsoft.com/en-us/sysinternals/bb842062

[h] Windows Server 2016 new Current Branch for Business servicing option
https://blogs.technet.microsoft.com/windowsserver/2016/07/12/windows-server-2016-new-current-branch-for-business-servicing-option/

[i] Software Assurance overview
https://www.microsoft.com/en-us/licensing/licensing-programs/software-assurance-default.aspx

Fazit

Für das Hosting von Fileservern, virtuellen Systemen, Webanwendungen und Webservices sowie Container-basierten Anwendungen ist der Nano Server ein interessantes Konzept, das das Potential hat, sich durch die Reduktion auf das Wesentliche viele Freunde bei den Administratoren zu machen, insbesondere auch beim Betrieb in Docker-Containern. Die Beschränkung auf .NET Core und einige Windows-APIs schränkt das Einsatzgebiet von Nano Server für den Betrieb von Anwendungen aber vorerst erheblich ein, denn es gibt noch keine Anwendungen für das kürzlich erschienene .NET Core und ASP.NET Core.

Die webbasierten Server Management Tools schrecken – gerade in Deutschland – noch Kunden ab, die sich mit der Cloud nicht anfreunden können – insbesondere in Anbetracht der Tatsache, dass das Administrator-Kennwort für ihre Windows Server in einem Webportal von Microsoft eingegeben werden soll. Nano Server ist durch das Support-Modell keine Option für konservativere IT-Abteilungen, sondern richtet sich klar an die agile Fraktion. In einem Blogeintrag [i] weist Microsoft zudem darauf hin, dass für den produktiven Einsatz von Nano Server ein Software-Assurance-Vertrag notwendig ist, was wiederum kleinere Unternehmen am Einsatz von Nano Server hindern wird. (jab)

Literatur

[1] Holger Schwichtenberg, Thomas Wiefel; Systemmanagement; Perlenfischer; Windows und Linux verwalten mit Microsofts Desired State Configuration; *iX* 5/2016, S. 64

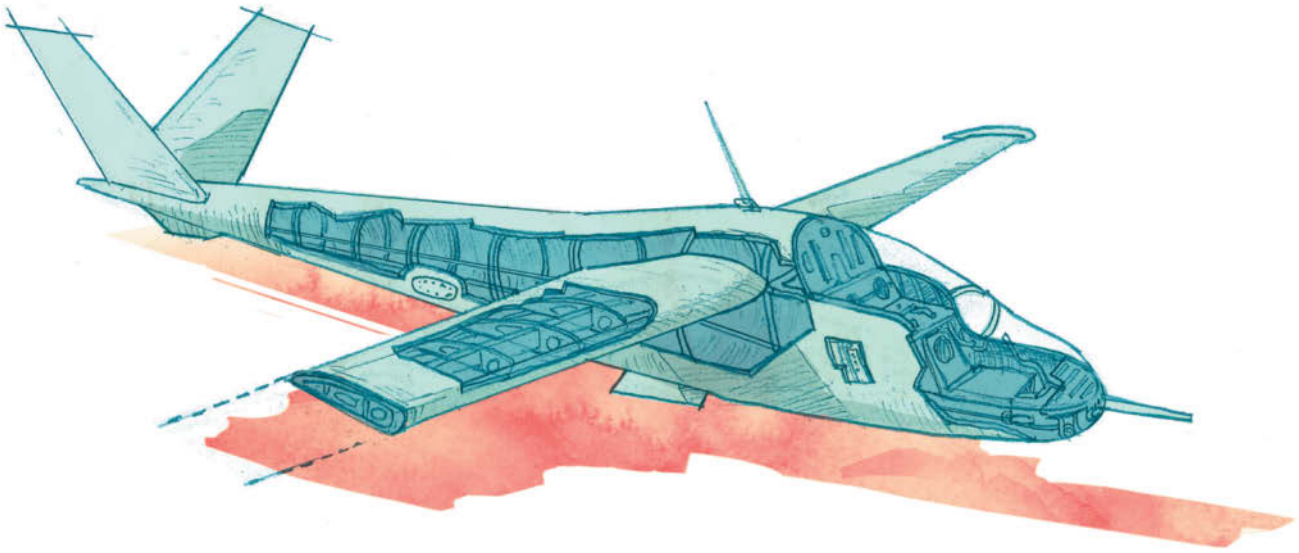
Dr. Holger Schwichtenberg
leitet das Expertennetzwerk www.IT-Visions.de, das Beratung, Schulungen und Softwareentwicklung im Umfeld von Microsoft-Technik anbietet.

Alle Links: www.ix.de/ix1616008

Drei Minimal-Betriebssysteme für Cloud-Dienste

Kompakt abheben

Udo Seidel



Auch im Linux-Umfeld hat die omnipräsente Cloud ihre Spuren hinterlassen, und das auf vielfältige Weise. Einer dieser „Effekte“ sind neue quelloffene Cloud-Betriebssysteme. Deren Ansätze sind jedoch keineswegs homogen und erfordern teilweise drastische Veränderung im Betrieb sowie in der Anwendungsentwicklung. Drei Kandidaten zeigen das ganze Spektrum.

Die Anforderungen an ein Betriebssystem in der Cloud sind nicht wirklich vergleichbar mit denen an ihr Pendant in einem klassischen Rechenzentrum. So ergibt etwa die Verfügbarkeit von Patches über einen Zeitraum von 10 Jahren oder länger im Cloud-Umfeld kaum Sinn. Die Lebenszeit eines Linux-Systems in der Cloud spielt sich mitunter im Rahmen von Stunden, Tagen oder bestenfalls Wochen ab. Themen wie Patching oder Updates erfordern ebenfalls neue Denkansätze. Statt der Zusammenarbeit mit beliebiger Hardware steht die reibungsfreie Interaktion mit der jeweiligen Cloud-Infrastruktur im Vordergrund. Genau genommen rückt das Betriebssystem selbst in den Hintergrund und ist nur das Vehikel für die Anwendung, die darauf läuft. Am Ende bleibt die Forderung nach einem System, das die guten Eigenschaften der vergangenen 25 Jahre mit der „neuen Mode“ des Cloud-Zeitalters vereint. Das bedeutet: alte Zöpfe abschneiden, unnötigen Zusatzaufwand über Bord werfen. Der Artikel beschäftigt sich mit drei Vertretern von Cloud-Betriebssystemen, die außerdem quelloffen sind. Das eben benutzte Bild aufgreifend, unterschieden sie sich darin, wie kurz sie die alten Zöpfe schneiden. An einem Ende des Spektrums steht das innovative CoreOS (siehe „Alle Links“ am Artikelende), am anderen Ende MirageOS, dazwischen liegt OSv.

CoreOS – Bekanntes und viel Neues

Die Wurzeln von CoreOS liegen in Googles Betriebssystem ChromeOS für seine Web-Notebooks. Die Erstveröffentlichung erfolgte im Herbst 2013 unter der Version 2 der Apache-Lizenz. Da CoreOS auf Komponenten wie dem Linux-Kernel und *systemd* aufbaut, gelten dort abweichende lizenzrechtliche Befugnisse. Die Eckdaten: Es ist zunächst ein abgespecktes Linux – der Verbrauch auf der (virtuellen) Festplatte und im Arbeitsspeicher liegen bei zirka 30 beziehungsweise 140 MByte.

CoreOS verzichtet auf einen Paketmanager, wofür es mehrere Argumente gibt. So muss das erwähnte Abspecken des Linux-Systems irgendwann zu Lasten der installierten Anwendungen gehen. Das Betriebssystem soll nur einige wenige Grundfunktionen erfüllen. Die Hauptarbeit verrichten die anderen Basiskomponenten, dazu gleich mehr. Zudem schwindet mit sinken-

der Anzahl der installierten Software auch die Notwendigkeit, diese mit einem Werkzeug zu verwalten. Als Konsequenz ist der Upgrade-Mechanismus von CoreOS deutlich „grober“ angelegt. Statt einzelner Dateien oder Verzeichnisse tauscht man hier komplette Dateisysteme aus: */usr* und */boot*. Die vorher genannte Hauptarbeit verrichten Linux-Container wie Docker beziehungsweise Rocket, etcd und fleet.

Die Linux-Container-Technik ist übrigens ein weiterer Grund, das darunterliegende Linux klein, ja geradezu minimalistisch zu halten. Alle auf CoreOS aufsetzenden Anwendungen sollen ohnehin in Containern laufen, inklusive der zusätzlich benötigten Software. etcd ist eine Art verteilter Datenträger für einfache Schlüssel-Wert-Paare mit eingebauter Cluster-Technik. Die ausführliche Dokumentation des CoreOS-Projektes spricht daher oft vom etcd-Cluster. fleet hingegen ist ein verteiltes *init*-System. Es setzt auf den *systemd*-Instanzen der einzelnen CoreOS-Rechner auf und erweitert sie zu einem Cluster. Dafür greift es auf die entsprechenden Funktionen von etcd zurück.

Versionen, Kanäle und der Kern

Technisch gesehen bestimmen also die Versionsnummern vom Linux-Kernel, von Docker oder Rocket, von etcd und von fleet eine CoreOS-Release. Tatsächlich sieht das Projekt dies zwar ähnlich, aber doch anders: Die Versionsnummer entspricht der Anzahl von Tagen, die seit Beginn der CoreOS-Epoche – dem 1. Juli 2013 – vergangen sind. Die erste Version erschien am 3. Oktober 2013 und trug folglich die Nummer 94. Genau genommen halten sich die Entwickler an die sogenannte semantische Versionierung, die erste Version war also 94.0.0. Neue Funktionen oder Fehlerkorrekturen, die rückwärtskompatibel sind, führen somit nicht zur Erhöhung der „Tagesanzahl“. Zum Zeitpunkt des Entstehens dieses Artikels listet das Projekt die aktuellste stabile Version von CoreOS unter der Nummer 1068.10.0. Insgesamt beinhaltet das Entwicklungsmodell drei Kanäle: Neben Stable findet man Beta und Alpha. Die Parallelen zu Debian sind wohl nicht zufällig.

Fehlerkorrekturen oder neue CoreOS-Funktionen implementieren die Entwickler zunächst im Alpha-Kanal. Treten dort keine Fehler (mehr) auf, wandern diese Änderungen in den Beta-Zweig. Die Beförderung in den Stable-Kanal erfolgt auf gleiche Weise. Wie oben beschrieben, basiert CoreOS auf dem Linux-Kern. Ein Blick in die Protokoll-Datei des Git-Verzeichnisses offenbart keine Auffälligkeiten – es handelt sich um einen Standard-Kernel, der schlicht weniger Optionen aktiviert hat als beispielsweise ein übliches Desktop-Linux. Ein CoreOS-Kernel hat mehr als die Hälfte weniger Fähigkeiten (Features) als seine Gegenstücke von den üblichen Distributionen.

Mit aktiviertem etcd und fleet beträgt die Anzahl der gestarteten Prozesse ungefähr 20. Neben *systemd*, Docker, etcd und fleet fallen hier noch locksmithd und update-engine auf, die Bestandteile der Upgrade-Technik von CoreOS sind und Hand in Hand arbeiten. locksmithd hört übrigens offiziell auf den Namen Reboot Manager. Ein Blick auf die Partitionstabelle, die Dateisysteme und ihre Verwendung hält ein paar Überraschungen bereit (Abbildung 2): Schreibzugriff auf */usr* ist im Standardfall nicht vorgesehen. Um bestimmte Dateien in */etc* ebenfalls vor Veränderungen zu schützen, sind diese durch symbolische Links auf entsprechende Gegenparts in */usr* ersetzt.

Auch ein derart verschlanktes System soll an die eigenen Bedürfnisse angepasst werden. Sowohl Benutzer als auch OEM-Distributoren nutzen dafür das Cloud-Config-Konzept. Mit YAML (YAML Ain’t Markup Language) als Beschreibungssprache hinterlegt der Anwender seine Anweisungen in einer Datei. Für die OEM-Distributoren ist sogar eine eigene Partition vorgesehen. Man findet sie nach dem Booten unter */usr/share/oem* eingehängt. Liegt dort eine Datei *cloud-config.yml*, integriert CoreOS diese in den Boot-Prozess und arbeitet sie ab.

Listing 1: Einfache Cloud-Config-Datei für eine CoreOS-Instanz

```
ssh_authorized_keys:
  - ssh-rsa AAAAB4NzaC1yC2dccccdjWFFFQEFyv...

hostname: coreos1.meinedomaene.de

users:
  - name: userA
    passwd: $6$SALT$qIet5289...
    groups:
      - sudo
      - docker
    ssh-authorized-keys:
      - ssh-rsa AAAAB3NzaC1YC2cfffabiwAAAQE...
$
```

```
$ tail -8 hello.cc

#include <iostream>

int main()
{
    std::cout << "Hi iX this is an OSv instance" << std::endl;
}

$
$ ~/bin/capstan run
Created instance: capstan-ix
OSv v0.24
eth0: 192.168.122.15
Hi iX this is an OSv instance

$
```

Die „Hello World“-Variante für OSv. Die auszuführende Anwendung liegt als *.so*-Datei vor (Abb. 1.)

Selbst Hand anlegen

Für den normalen Anwender stellen die Entwickler den von Openstack bekannten Config-Drive-Ansatz bereit. Die Cloud-Config-Datei liegt auf einem Dateisystem, das CoreOS beim Hochfahren einbindet und verarbeitet. Zur Laufzeit der CoreOS-Instanz findet man diese Datei im Verzeichnis */media/configvirtfs/openstack/latest* (siehe Listing 1). Eine Anleitung für die Integration in die verschiedenen Cloud-Plattformen findet sich unter coreos.com/docs/#running-coreos.

Getreu dem minimalistischen Ansatz kommt CoreOS nur mit zwei Benutzer-Accounts daher: *root* und *core*. Für beide ist eine interaktive Anmeldung ohne Zusatzaufwand nicht vorgesehen. Es gilt, SSH-Schlüssel anzulegen und den öffentlichen Teil über die Cloud-Config-Datei zu integrieren (ebenfalls in Listing 1). Ein ähnliches Vorgehen ist für das Anpassen des Rechnernamens, das Anlegen weiterer Benutzer und das Erzeugen beziehungsweise Aktualisieren von Dateien vorgesehen.

Für die ersten eigenen Schritte mit CoreOS gibt es viele Möglichkeiten: Die Projekt-Dokumentation listet ungefähr 20 verschiedene Optionen. Da sind einmal kommerzielle Cloud-Anbieter wie Amazon Web Services, Microsoft Azure, Google Compute Engine oder der Open-Source-Star Openstack. Es geht jedoch auch ohne Cloud: Mit Qemu, livbvirt, VirtualBox und VMware sind die Anfänge in der heimischen IT ebenfalls recht einfach. Die Images, auch die der offiziellen Cloud-Anbieter, kann man sich unter

```
http://<Kanal>.release.core-os.net
```

herunterladen. Dabei entspricht der <Kanal> dem Entwicklungszweig, ist also entweder Stable, Beta oder Alpha. So oder so: CoreOS ist um einiges anders als die traditionellen Linux-Distributionen. Für ein vollständiges Bild sollte man sich zudem

Wichtige *capstan*-Kommandos zur Verwaltung von OSv-Abbildern und -Instanzen

Kommando	Beschreibung
pull	Lädt ein OSv-Abbild vom Github-Repository
rmi	Löscht ein OSv-Abbild im lokalen Repository
run	Startet eine OSv-Instanz
build	Generiert ein OSv-Abbild
images	Listet die lokalen OSv-Abbilder auf
search	Sucht nach OSv-Abbildern im Github-Repository
instances	Listet lokale OSv-Instanzen auf
stop	Stoppt eine OSv-Instanz
delete	Löscht eine OSv-Instanz

ausführlich mit etcd und fleet befassen (aus Platzgründen nicht im Artikel enthalten). Eine kurze Einführung liefert [1].

OSv – Einen Gang höher schalten

Das OSv-Projekt ist nicht wirklich älter als CoreOS. Die erste Version kam im September 2013 heraus. Ein Blick in den Git-Log zeigt, dass der erste Eintrag am 1. Dezember 2012 erfolgte. Auf der LinuxCon Europe 2013 in Edinburgh war OSv schon ein kleiner Star. Dieses eigens für die Cloud entwickelte Betriebssystem zog das Interesse der Linux-Gemeinde auf sich. Cloudius Systems, das Team hinter OSv, stammt aus der Linux-Hypervisor-Szene und ist kein Unbekannter. Das Linux-artige Betriebssystem versteht sich als minimalistische Schicht zwischen Hypervisor und der Cloud-Anwendung. Installation und Betrieb auf echter Hardware ist nicht vorgesehen. Beim Hypervisor hat der Interessent die Wahl zwischen KVM, Xen, VMware und VirtualBox. Selbstverständlich funktioniert OSv auch in den bekannten öffentlichen Cloud-Umgebungen. Der Anwender hat beispielsweise die Wahl zwischen Amazons Elastic Cloud (EC2) oder Googles Compute Engine (GCE).

Die grundlegende Design-Idee der Entwickler ist eine radikale Reduzierung der Betriebssystemschicht. Diese muss nur zwei Aufgaben erfüllen: Einerseits das Zusammenspiel mit nahezu beliebigen Hypervisors sowie die Bereitstellung der Plattform für die eigentliche Cloud-Applikation. Auch hier gilt das Minimalismus-Prinzip – die Laufzeitumgebung für eine POSIX-kompatible C++-Anwendung muss nicht besonders groß sein. Weitere Aufgaben hat das Betriebssystem nicht zu verrichten. Gleiches gilt für Java oder Ruby. Mit diesem Ansatz lässt sich ein herkömmliches System schon stark verschlanken. Die Entwickler sind jedoch noch einen Schritt weiter gegangen und haben OSv von Grund auf neu geschrieben. Im Gegensatz zum traditionellen Linux ist C++ die Programmiersprache der Wahl. OSv steht zu großen Teilen unter der 3-Clause-BSD-Lizenz. Verschiedene andere Open-Source-Projekte standen Pate für den Programm-Code, etwa FreeBSD, OpenSolaris, Prex und Musl. Damit gelten für die entsprechenden OSv-Bestandteile die jeweiligen Lizenzen des Ursprungs-Quelltextes.

Bei OSv laufen alle Prozesse im Kernel-Space. Das hat zwei Konsequenzen: Das System ist schlanker und schneller, und ein Nutzerkonzept ist hinfällig. Ein Kontextwechsel zwischen Kernel- und Userspace fällt weg, ebenso das Umschalten zwischen Applikationen. Per Design führt OSv nur eine einzige Anwendung – streng genommen nur einen Prozess – aus. Davon profitieren beispielsweise die Laufzeitumgebung für Java-Anwendungen. Die Java Virtual Machine (JVM) ist gewissermaßen ein Bestandteil des Kernels.

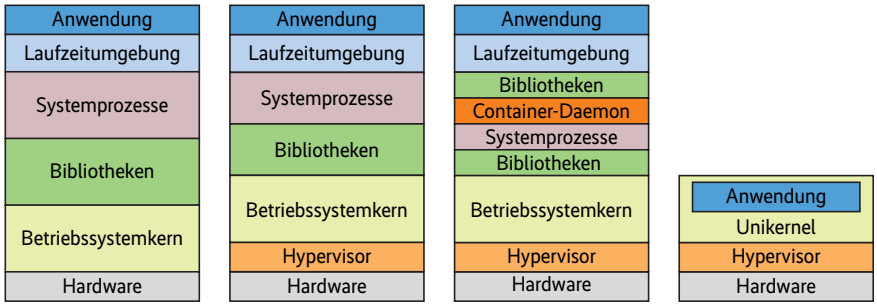
OSv reicht Teile der Speicherverwaltung direkt weiter. So muss die JVM nicht die traditionellen Schnittstellen wie *mmap()* benutzen. Damit noch nicht genug: Das Betriebssystem verzichtet komplett auf Spinlocks. Die Entwickler versuchen die Entwicklung von Locks generell zu vermeiden. Wo dies nicht möglich ist, kommt eine spezielle Mutex-Variante zum Einsatz. Dieses Konzept zum Vermeiden von Locks findet sich auch im Netz-Stack: Hier setzt OSv auf das Kanalkonzept von Van Jacobson (siehe „Alle Links“). Aus dem FreeBSD-Umfeld haben die Entwickler die Datenträgerverwaltung übernommen. Es überrascht daher nicht, dass man hier das Dateisystem ZFS vorfindet. Das Thema Sicherheit haben die OSv-Entwickler komplett an das darunterliegende Hostsystem delegiert. Das Resultat ist schließlich ein Betriebssystem, das nur 30 MByte groß ist. Zum Vergleich: Ein „handelsüblicher“ Desktop-Linux-Kernel belegt ohne Weiteres mehr als 6 MByte. OSv ist ein 64-Bit-Betriebssystem und primär in der x86-Welt zu Hause. Die Umsetzung für ARM zeigt jedoch schon erste Ergebnisse.

Noch mehr als CoreOS ist OSv anders als ein traditionelles Linux (oder Free/Net/OpenBSD). Beim Benutzen dieses Systems gibt es zwei Aspekte zu beachten: Der Administrator stellt meist die Frage, wie er ein System verwalten soll, das keine Benutzer kennt, bei dem alles im Kernel-Space läuft und das nur für eine einzelne Anwendung ausgelegt ist. Der Anwender wiederum möchte wissen, wie er seine Lieblingsanwendung auf OSv zum Laufen bekommt. Für Letzteres verfügt es über die meisten Linux-ABIs (Application Binary Interface) – unter Berücksichtigung der Design-bedingten Einschränkung.

Wegen des nicht vorhandenen Mehrbenutzer-Konzepts entfallen Systemaufrufe wie *fork()*, *vfork()* und *clone()* ersatzlos –

```
udo@coreos3 ~ $ grep vda /proc/mounts
/dev/vda9 / ext4 rw,seclabel,relatime,data=ordered 0 0
/dev/vda3 /usr ext4 ro,seclabel,relatime,block_validity,delalloc,barrier,user_xattr,acl 0 0
/dev/vda6 /usr/share/oem ext4 rw,seclabel,nodev,relatime,commit=600,data=ordered 0 0
/dev/vda1 /boot vfat rw,relatime,fmask=0022,dmask=0022,codepage=437,iocharset=ascii,shortname=mixed,errors=remount-ro 0 0
udo@coreos3 ~ $ ls -l /etc/lgrep ^l
lrwxrwxrwx. 1 root root          36 Aug 23 02:27 idmapd.conf -> ../usr/share/libnfsidmap/idmapd.conf
lrwxrwxrwx. 1 root root          31 Aug 23 02:26 inputrc -> ../usr/share/baselayout/inputrc
lrwxrwxrwx. 1 root root          12 Aug 23 02:30 issue -> ../run/issue
lrwxrwxrwx. 1 root root          21 Aug 23 02:30 ld.so.conf -> ../usr/lib/ld.so.conf
lrwxrwxrwx. 1 root root          26 Sep  5 18:04 limits -> ../usr/share/shadow/limits
lrwxrwxrwx. 1 root root          25 Sep  5 18:04 localtime -> ../usr/share/zoneinfo/UTC
lrwxrwxrwx. 1 root root          32 Sep  5 18:04 login.access -> ../usr/share/shadow/login.access
lrwxrwxrwx. 1 root root          30 Aug 23 02:29 login.defs -> ../usr/share/shadow/login.defs
lrwxrwxrwx. 1 root root          31 Aug 23 02:26 lsb-release -> ../usr/share/coreos/lsb-release
lrwxrwxrwx. 1 root root          18 Aug 23 02:26 motd -> ../run/coreos/motd
lrwxrwxrwx. 1 root root          19 Sep  5 18:04 mtab -> ../proc/self/mounts
lrwxrwxrwx. 1 root root          37 Aug 23 02:26 nsswitch.conf -> ../usr/share/baselayout/nsswitch.conf
lrwxrwxrwx. 1 root root          23 Sep  5 18:04 ntp.conf -> /usr/share/ntp/ntp.conf
lrwxrwxrwx. 1 root root          21 Aug 23 02:26 os-release -> ../usr/lib/os-release
lrwxrwxrwx. 1 root root          31 Aug 23 02:26 profile -> ../usr/share/baselayout/profile
lrwxrwxrwx. 1 root root          38 Aug 23 02:27 request-key.conf -> ../usr/share/keyutils/request-key.conf
lrwxrwxrwx. 1 root root          34 Sep  5 18:04 resolv.conf -> ../run/systemd/resolve/resolv.conf
lrwxrwxrwx. 1 root root          29 Aug 23 02:29 securetty -> ../usr/share/shadow/securetty
lrwxrwxrwx. 1 root root          30 Aug 23 02:26 shells -> ../usr/share/baselayout/shells
lrwxrwxrwx. 1 root root          33 Aug 23 02:26 sudo.conf -> ../usr/share/baselayout/sudo.conf
udo@coreos3 ~ $
```

CoreOS sieht im Normalbetrieb keine Änderungen an */usr* vor – dies gilt sogar für einige Dateien in */etc*, die lediglich Symlinks darstellen (Abb. 2).



Der Unikernel-Ansatz (rechts) im Vergleich mit einem herkömmlichen Betriebssystem (links), einem mit Hypervisor (zweites von links) sowie mit Containern (zweites von rechts, Abb. 4).

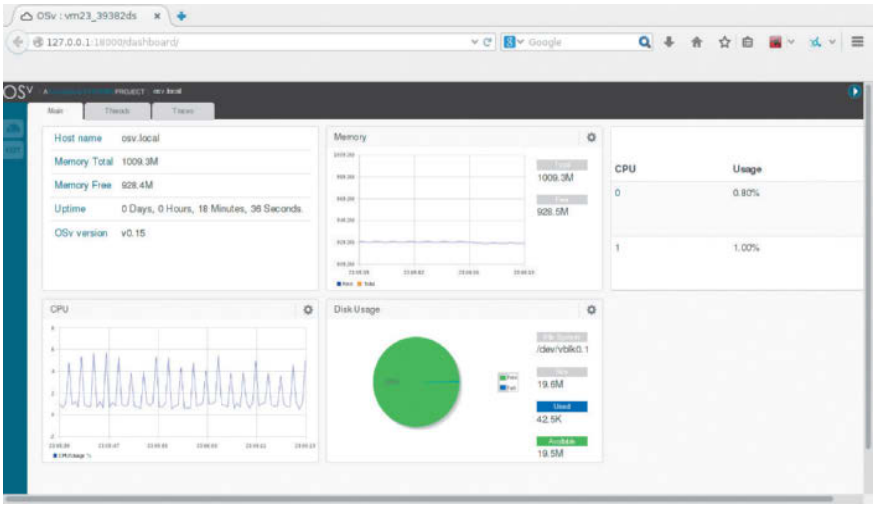
schliesslich läuft nur ein Prozess. Dazu kommt, dass die auszuführende Anwendung im sogenannten „relocatable shared object"-Format vorliegen muss. OSv sucht im Objekt-Code nach der Hauptfunktion *main()* und führt diese aus. Dem Linux-Anwender begegnet dieses ständig in Form der Bibliotheken seines Systems. Abbildung 1 zeigt eine Variante des berühmten „Hello World"-Beispiels für OSv.

Die Anwendung hinterlegt man in einem OSv-Image im Verzeichnis */tools*. Anschließend muss man das Image noch so konfigurieren, dass es die Anwendung beim Start ausführt. Das Resultat ist eine OSv-Instanz (Abbild inklusive Anwendung). Dieses Verfahren zeigt deutliche Parallelen zum sogenannten Unikernel-Ansatz. Tatsächlich nennen neuere Dokumente OSv immer wieder in diesem Zusammenhang. Zum Thema Unikernel gibt es weiter unten mehr Informationen. Das Zusammenbauen einer fertigen OSv-Instanz inklusive Anwendung ist im Detail nicht ganz einfach. Schon das Ablegen der Anwendung könnte an der fehlenden Unterstützung für die richtige ZFS-Version scheitern. Alternative Ansätze wie der Befehl *scp* scheitern am Minimalismus des Betriebssystems – dort läuft kein SSH-Daemon. Zur Verwaltung benutzt der OSv-Administrator das Werkzeug Capstan. Wie das Betriebssystem steht es unter der 3-Clause-BSD-Lizenz. Die wichtigsten Kommandos führt Tabelle 1 auf. Eine gewisse Ähnlichkeit zur Verwaltung von Linux-Containern mit dem Kommandozeilenwerkzeug Docker lässt sich nicht leugnen.

Selbst mit Capstan kann das Erzeugen des OSv-Abbilds mit der gewünschten Anwendung aufwendig sein. Glücklicherweise existiert eine Art Marktplatz mit vorgefertigten Applikationen (osv.io/downloads/). Für Java-Anwendungen gibt es unter OSv lauffähige JVMs mit den openJDK-Versionen 7 und 8. Der Start der zugehörigen Anwendung sieht ungefähr so aus:

```
/tools/java.so -jar app.jar
```

Wer tiefer einsteigen will, findet auf der Wiki-Seite des Projekts die gewünschten Informationen. Und wer mehr über den aktuellen Zustand seiner OSv-Instanzen wissen muss, kann beispielsweise das eingebaute Dashboard benutzen, das über HTTP auf Port 8000 zur Verfügung steht (Abbildung 3). Der traditionelle Linux-Anwender findet eine – wenn auch stark eingeschränkte – Shell vor. Der empfohlene Weg ist jedoch das Verwenden der REST-API, die ebenfalls in einem Wiki ausführlich dokumentiert ist.



OSv hat einen kompakten Webserver mit Dashboard eingebaut – zum Auslesen des „Gesundheitszustandes" der Instanz (Abb. 3).

MirageOS – ein radikaler Ansatz

Das Projekt MirageOS ist deutlich älter als die beiden zuvor behandelten Projekte. Laut Aussagen der Entwickler reichen die Anfänge bis ins Jahr 2009 zurück. An die breite Öffentlichkeit trat das Projekt mit Version 1.0 allerdings erst im Dezember 2013. Ähnlich wie OSv geht dieses Cloud-Betriebssystem einen ganz neuen Weg. Genau genommen versteht sich MirageOS gar nicht als Betriebssystem, sondern eher als eine Art Bibliothek. Die Grundidee ist dabei noch radikaler als bei OSv: Hier läuft die Anwendung im Kernel-Space und definiert zusammen mit dem Betriebssystemkern quasi den kompletten virtuellen Server. Viele Aufgaben eines traditionellen Kernels fallen weg oder sind deutlich reduziert. Dazu gehört die Verwaltung des Netzes und des Hauptspeichers. Das eröffnet zudem neue Möglichkeiten bezüglich der Auswahl der Programmiersprache – dazu später mehr.

Vereinfacht gesagt ist MirageOS eine Art Baukasten zum Erzeugen eines eigenen Betriebssystems. Dieses ist auf den Kernel reduziert, der zugleich die Anwendung ist. Dieser Ansatz hat seit ein paar Monaten wieder erhöhte Aufmerksamkeit erfahren und ist unter dem Namen Unikernel bekannt. Die Evolution der herkömmlichen Betriebssysteme und ein Vergleich mit dem eben genannten Konzept ist in Abbildung 4 dargestellt. Neben MirageOS gibt es eine ganze Reihe von Projekten für und mit Unikernels (unikernel.org/projects/). Selbst die Fans der Linux-Container haben Interesse an dem Konzept gefunden.

MirageOS steht unter der ISC-Lizenz und ist organisatorisch als Inkubator-Projekt innerhalb von Xen gelistet. Die x86-Architektur zählt selbstverständlich zu den unterstützten Plattformen der ersten Stunde. Mit Version 2.0 vom Juli 2014 kam ARM dazu. Will man MirageOS benutzen, muss man sich mit der Programmiersprache OCaml befassen. Für die als singuläre Applikation funktionierenden Kerne – also Unikernels – gibt es eine Tendenz zur Verwendung von Programmiersprachen, die nicht so „nah" an der Hardware oder Technik arbeiten. Wie bereits angedeutet, ergeben sich daraus einige Vorteile: Im Fall von OCaml nennen die Entwickler automatische Speicherverwaltung, Typenprüfung während des Kompilierens und modularen