



## Microsoft Azure 自習書シリーズ コマンドラインによる仮想マシンの構成と操作

---

この自習書では、Azure PowerShell を使用し、Azure 仮想マシンおよび、仮想マシンの関連リソースを作成する方法を学習します。

発行日 : 2015 年 12 月 21 日

## 更新履歴

---

版数	発行日	更新履歴
第 1 版	2015 年 12 月 21 日	初版発行

## 目次

---

Microsoft Azure 自習書シリーズ コマンドラインによる仮想マシンの構成と操作 .....	1
更新履歴 .....	2
目次.....	3
1. はじめに.....	5
1.1 前提知識.....	5
1.2 コマンドライン ツール.....	5
1.3 Azure の 2 種類のデプロイメント モード.....	6
1.4 Azure PowerShell .....	6
1.5 Windows PowerShell ISE .....	7
1.6 本自習書のシナリオと目標.....	8
1.7 Azure サービスのプロビジョニング モデル .....	9
2. 事前準備.....	10
2.1 Microsoft Azure 無料評価版のサインアップ手順 .....	12
2.2 Azure PowerShell 1.0.1 モジュールのインストール手順.....	15
2.3 認証とサブスクリプションの選択.....	20
3. PowerShell による IaaS v2 リソースの構築 .....	26
3.1 リソース グループの作成.....	27
3.2 ストレージ アカウントの作成.....	33
3.3 仮想ネットワークとサブネットの作成 .....	37
3.4 ネットワーク インターフェイスとパブリック IP アドレスの作成.....	49
3.5 仮想マシンの作成 .....	58
3.6 カスタム スクリプト拡張を使用した IIS 機能のインストール .....	72

3.7	作成したリソース グループの削除.....	77
4.	PowerShell と JSON テンプレートによる IaaS v2 リソースの構築.....	78
4.1	リソース グループの作成.....	79
4.2	ストレージ アカウントの作成.....	80
4.3	仮想ネットワークとサブネットの作成 .....	82
4.4	ネットワーク インターフェイスとパブリック IP アドレスの作成.....	84
4.5	Azure 仮想マシンの作成.....	90
4.6	DSC 拡張機能を使用した IIS 機能のインストール.....	98
4.7	仮想マシンの停止 .....	105
4.8	作成したリソース グループの削除.....	106
4.	Microsoft Azure に関する情報の入手元.....	107
5.	Microsoft Azure のお問合せ .....	108

## 1. はじめに

---

本自習書をご利用いただきありがとうございます。本書では、Azure PowerShell 1.0.1 を使用して Web サーバーのための仮想マシンと SQL Server が稼働するデータベース サーバーのための仮想マシンの構築しながら、以下の項目を学習することができます。

- Windows PowerShell ISE を使用した Azure PowerShell スクリプトの作成方法
- Azure PowerShell 1.0.1 を使用して ARM モードで、IaaS v2 の Azure リソースを作成する方法
- コマンドレットで使用するパラメーターを変数化することで、スクリプトの再利用性を向上させる方法
- JSON テンプレートとパラメーター ファイルを使用した IaaS v2 リソースのプロビジョニング方法
- Azure 仮想マシンのカスタム スクリプト拡張機能を使用して IIS をインストールする方法
- Azure 仮想マシンの DCS 拡張機能を使用して IIS をインストールする方法

この自習書は、ARM モードで、IaaS v2 リソースを作成するための Azure PowerShell コマンドレットの使用方法を学習します。2 章で実習環境を作成し、3 章では、Azure PowerShell の各コマンドレットを使用しながらストレージ、仮想ネットワーク、仮想マシンを作成します。4 章では、3 章で作成したリソースと同じものを JSON テンプレートに宣言的に定義し、New-AzureResourceGroupTemplate コマンドレットで ARM デプロイメント モードの IaaS v2 リソースを作成します。

なお、本自習書では、仮想マシンのスケールアウト構成で使用するロード バランサーの構成を Azure PowerShell で実装する方法は記載しておりません。このトピックスについては、「Microsoft Azure を活用したスケール実現方法」の「4. Azure 仮想マシンによるスケーラブルな Web サイトの構築」を参照してください。

### 1.1 前提知識

この自習書で実習を行うために必要な知識は以下のとおりです。

- Microsoft Azure 仮想マシンを作成したことがある方、または同程度の知識
- PowerShell に関する基本的な知識と操作方法

まだ、Azure 仮想マシンを作成したことがない場合は、自習書「Microsoft Azure 自習書シリーズ 仮想マシンの作成と操作」から学習されることをお勧めします。

### 1.2 コマンドライン ツール

Windows PowerShell に Azure PowerShell モジュールをインストールすることで、Azure を管理するコマンドレットを使用することができます。Azure PowerShell を使用して、Azure プラットフォームが提供するサービスを作成、テスト、配置、および、管理することができます。Azure PowerShell は、コマンド プロンプトで対話形式によって操作する方法でも、スクリプトでバッチ処理する方法でも使用できます。コマンドレットを使用して、Azure 管理ポータルで実行できるタスクと同じタスクを実行できます。例えば、リソース グループを作成し、その中に、関連するストレージ アカウント、仮想ネットワーク、仮想マシンなどを含めた様々な Azure リソースを作成することができます。

### 1.3 Azure の 2 種類のデプロイメント モード

Azure IaaS サービスでは、Azure Service Management (ASM) と Azure Resource Management (ARM) の 2 種類のデプロイメント モードが選択できます。

- Azure Service Management (ASM)

プレビュー ポータルでは、クラシック デプロイ モデルと表現されている、従来からの クラウド サービスをコンテナとして使用するデプロイ モデルで、「仮想マシン v1」または、「IaaS v1」とも表現されています。ASM モードに配置されたりソースは、クラシック ポータルと新ポータルの両方から管理することができます。

- Azure Resource Management (ARM)

リソース マネージャーを使用した新しい仮想マシンで、「仮想マシン v2」、または、「IaaS v2」とも呼ばれます。新ポータルから操作することができます。

2 種類のデプロイメント モードの違いは、下表を参考にしてください。

仮想マシンのデプロイメント モード		
機能	ASM	ARM
管理方法	クラシックポータル、 プレビューポータル	プレビューポータルのみ
設定	XML ベース	JSON ベース
仮想ネットワーク	オプション	必須
ドメイン名	<クラウドサービス名>. cloudapp.net	<ドメインラベル名>.<場所>. cloudapp.azure.com
クラウドサービス	有	なし
アフィニティグループ	有	なし
可用性セットの最大障害ドメイン数	2	3
ファイアウォール機能	エンドポイント、ネットワーク セキュリティグループ	ネットワークセキュリティ グループ

### 1.4 Azure PowerShell

バージョン 0.9.8 より以前の Azure PowerShell では、2 種類のデプロイメント モード (ASM と ARM) で同じ名前の異なる動作をするコマンドレットがありました。例えば、ASM モードで Get-AzureVM コマンドレットを実行した場合は、v1 の 仮想マシンが一覧表示され、ARM モードで実行した場合は、v2 の 仮想マシンが一覧表示されます。Azure PowerShell では、0.9.8 まで、この環境を提供するため、2 種類のデプロイメント モードを切り替える Switch-AzureMode コマンドレットが使用されてきました。この環境切り替え用のコマンドレットを使用することで、選択されたデプロイメント モードのコマンドレットが実行されるようにモジュール パスを変更されます。しかし、これにより、モード変更前に使用していたコマンドレットはパスから削除され、使用できなくなってしまうといった不便があり、切り替えが必要なプロビジョニングでは、対話形式のみの使用方法に限定されてきました。

2015 年 10 月より提供が開始された Azure PowerShell 1.0 Preview で、この Switch-AzureMode コマンドレットは廃止され、ASM と ARM のコマンドレットを切り替えることなく使用できるように変更されました。このため、Azure PowerShell 0.9.8 の ARM モードで使用されていた一連のコマンドレットは、**[動詞]-Azure[名詞]** から **[動詞]-AzureRM[名詞]** に変更されています。

例えば、Get-AzureVM コマンドレットは、ARM モードで使用する場合、Get-AzureRMVM コマンドレットになります。なお、ASM モードのコマンドレットは、従来のもので、変更されていないため、これまで記述した ASM モードのスク립トは、Azure PowerShell 1.0 以降でも、そのまま使用することができます。

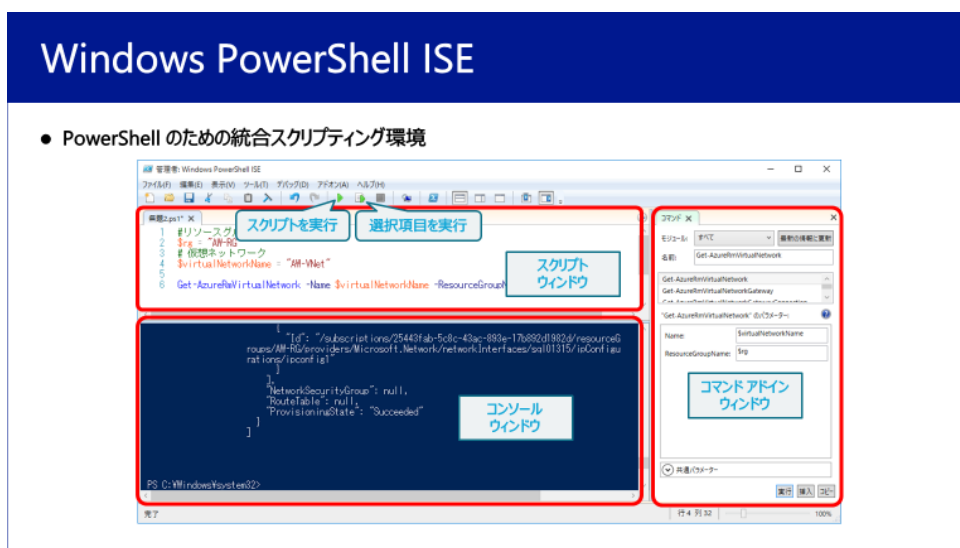
なお、本自習書では、Windows 10 を実行しているコンピューターに GA 済みの Azure PowerShell 1.0.1 のモジュールをインストールして、AzureResourceManager (ARM) モードで、Azure PowerShell スクリプトを実行しています。この自習書を実行される場合、同じバージョンの Azure PowerShell を使用するようにはしてください。実習環境の構築方法については、「2.2 Azure PowerShell 1.0.1 モジュールのインストール手順」を参照してください。

## 1.5 Windows PowerShell ISE

Windows PowerShell Integrated Scripting Environment (ISE) は、Windows PowerShell のホストアプリケーションで、GUI を使用してスクリプトの作成、テスト、およびデバッグ実行をすることができます。

スクリプト ウィンドウを使用して、スクリプトの複数行の編集、タブ補完、構文の色分け、スクリプト全体の実行、選択的な実行などの操作を行えるため、コンソールで行うより、効率よく作業することができます。また、コマンド アドイン ウィンドウを使用することで、使用するコマンドレットに必要なパラメーターを調べて、パラメーター値をセットして、そのまま必要なスクリプト行を生成できるため、構文エラーを減らすことができます。

以下は、Windows PowerShell ISE の各ウィンドウの名称です。実習手順を始める前に、確認しておく、分かりやすいでしょう。



## 1.6 本自習書のシナリオと目標

この自習書で、あなたは、Adventure Works Cycles 社 (AW 社) に勤めている IT 管理者として作業します。AW 社は、架空の大規模な多国籍製造企業です。この企業は、北米、ヨーロッパ、およびアジアのマーケットを対象に、金属製自転車やカーボン製自転車の製造および販売を行っています。従業員 290 人の米国ワシントン州ボセルの拠点に加え、自社のマーケット基盤全体にわたって複数の地域販売チームを配置しています。

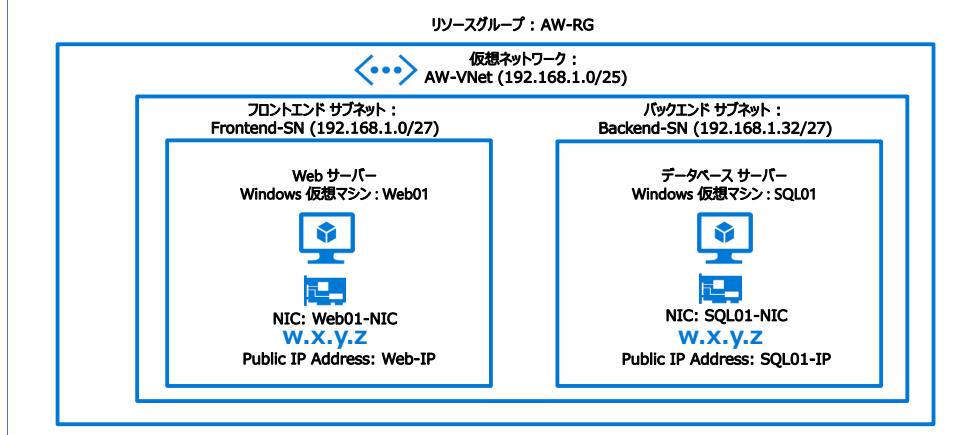
### Adventure Works Cycles とは

- 架空の企業
- この企業は、北米、ヨーロッパ、およびアジアのマーケットを対象に、金属製やカーボン製の自転車の製造および販売を行っている
- 従業員 290 人の米国ワシントン州ボセルの拠点に加え、自社のマーケット基盤全体にわたって複数の地域販売チームを展開



あなたは、Microsoft Azure の仮想マシンを利用し、営業部向けの Web システムをテスト運用するため、そのインフラを構築しなければなりません。この Web システムは、Windows Server 2012 R2 で実行される IIS Web サーバーと SQL Server 2014 SP1 が稼働するデータベース サーバーの 2 台で構成する必要があります。

### PowerShell で構築するシステム



あなたは、PowerShell を使用した Microsoft Azure プラットフォームに仮想マシンを作成するスクリプト ファイルを作成したいと考えています。この自習書では、Azure IaaS サービスに構築する Web サーバーと データベースサーバーのためのリソースは、リソース マネージャー (ARM) のデプロイメント モードで作成します。



## 1.7 Azure サービスのプロビジョニング モデル

### ● アカウントとサブスクリプション

Azure クラウド サービスに接続するためのアカウントとして、企業や教育機関の管理者が発行した組織情報、または、Microsoft アカウントを使用することができます。既に、Azure サービスを契約している組織では、ご自身が使用するアカウントが、共同管理者、またはサービス管理者の管理ロールとして設定されていれば、この自習書の手順をすべて実行することができます。

自習書の実行に使用できるアカウントが存在しない場合、ご自身が取得された Microsoft アカウントを Microsoft Azure サブスクリプションに割り当てることで、Azure サービスを利用することが可能になります。Microsoft アカウントは、マイクロソフトが提供する ID です。Microsoft Azure サブスクリプションは、Microsoft Azure を使用するための契約形態で無料評価版、従量課金、MSDN Platforms、Microsoft Online Subscription Program (MOSP)、OPEN、Azure EA (SCE)、MPSA (Select Plus の置き換え) から選択することができます。初めて、Azure を使用される場合は、以降の手順で紹介する 1 か月間使用できる無料評価版のサブスクリプションを取得されることをお勧めします。

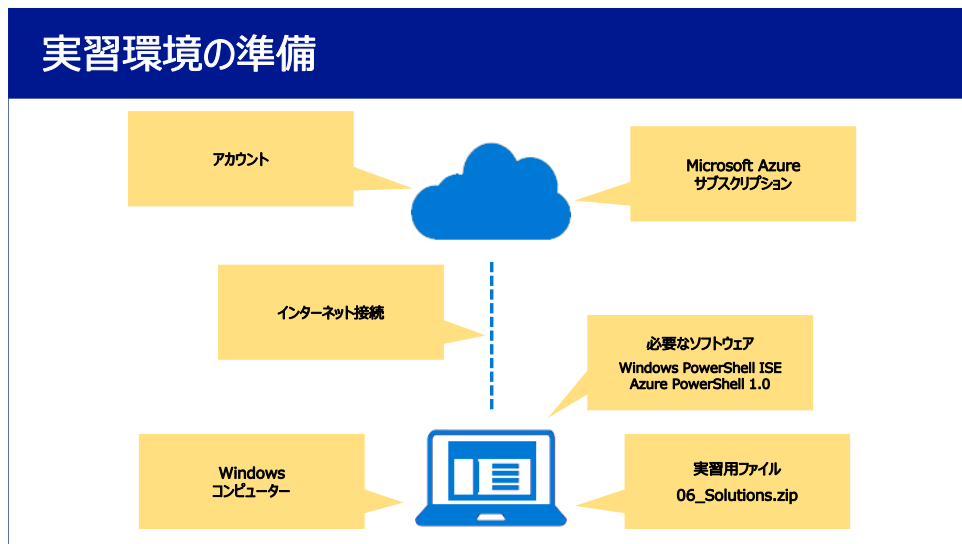
### Azure サービスのプロビジョニング モデル

- アカウント
  - 認証情報
  - Microsoft アカウント、組織アカウントから選択できる
- サブスクリプション
  - Azure サービス利用の契約形態
  - 無料評価版、従量課金、MSDN Platforms、Microsoft Online Subscription Program (MOSP) OPEN、Azure EA (SCE)、MPSA (Select Plus の置き換え) などがある
  - 無料評価版へのサインアップ ⇒ <http://azure.microsoft.com/ja-jp/pricing/free-trial/>
- Azure サービス
  - Microsoft Azure が提供する IaaS、および PaaS サービス
  - コンピューティング、データ、アプリケーション、ネットワークなどの各サービスに分類される

```
graph TD; A[アカウント] --> B[サブスクリプション]; A --> C[サブスクリプション]; B --> D[Azure サービス]; B --> E[Azure サービス]; C --> F[Azure サービス]; C --> G[Azure サービス];
```

## 2. 事前準備

この自習書の手順は、次の実習環境を準備することで、実際に試すことができ、理解を深めることができます。



- Windows コンピューター

実習用に使用する Windows 10 がインストールされた自由に操作できる Windows コンピューターが 1 台必要です。このコンピューターは、仮想マシンでも構いません。また、Azure ポータルに接続するための Web ブラウザーは、バージョン 11 以降の Internet Explorer、または、最新バージョンの Edge などを使用することができます。

- インターネット接続

Windows コンピューターは、インターネットに接続されている必要があります。企業内で実習をおこなう場合は特に注意が必要です。多く企業のインターネット接続では、ファイアウォールが介在します。その場合は実習で必要なリモート デスクトップ接続のためのポートや PowerShell 接続のためのポートがブロックされていないことを確認しておく必要があります。

- アカウントの取得

所属される企業で Azure サービスを作成できる権限が与えられた組織情報アカウントを入手できる場合は、その組織情報アカウントをご使用いただくか、または、Microsoft Azure サブスクリプションが割り当てられた Microsoft アカウントがあれば、それを使用することもできます。実習に使用できるアカウントをお持ちでない場合、Microsoft アカウントを <http://www.microsoft.com/ja-jp/msaccount/default.aspx> より、取得してください。

### ワンポイント

新しく、Microsoft Azure の 1 か月間の無料評価版のサブスクリプションをサインアップする際に、同時に Microsoft アカウントを登録することもできます。

- Microsoft Azure サブスクリプションの取得

使用するアカウントに、既に Microsoft Azure サブスクリプションが割り当てられている場合、そのサブスクリプションを利用することができます。Microsoft Azure サブスクリプションをお持ちでない場合、所有されている Microsoft アカウントを使用して、<https://azure.microsoft.com/ja-jp/pricing/free-trial/> より、無料評価版をサインアップすること

で、1 か月間 20,500 円相当のクレジットが利用可能です。無料評価版のサインアップ手順は、「2.1 Microsoft Azure 無料評価版のサインアップ手順」を参照してください。

#### ワンポイント

Microsoft Azure の 1 か月間の無料評価版のサインアップには、本人確認のため、電話番号（固定電話または携帯電話）およびクレジット カード情報が必要です。なお、過去に無料評価版のサインアップで使用した電話番号やクレジット カードはご利用できませんので、ご注意ください。

- 必要なソフトウェア

この自習書では、Windows PowerShell ISE を使用して、作成した PowerShell スクリプトでクラウド環境にリソースを作成する手順を実行します。実習用に使用する Windows コンピューターに、事前に Azure PowerShell 1.0 をインストールしておく必要があります。この手順は、「2.2 Azure PowerShell 1.0.1 モジュールのインストール手順」を確認してください。

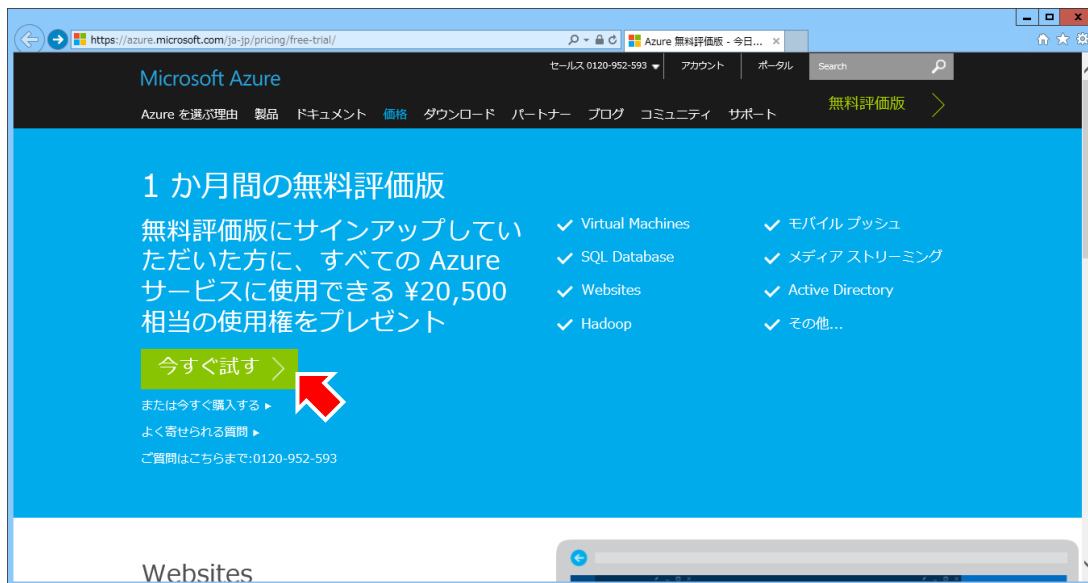
- 実習用ファイル

この自習書のサイトからダウンロードした **Solutions.zip** を取得してください。ZIP ファイルには、自習書で作成するすべてのスクリプト ファイルと JSON ファイルが含まれています。本自習書では、D ドライブに Solutions フォルダが開されているものとして記載していますが、ご自身の使用しているコンピューターのディスク環境に合わせて、展開先のフォルダを修正する場合、スクリプトに書かれた フォルダ パスも変更して使用ください。

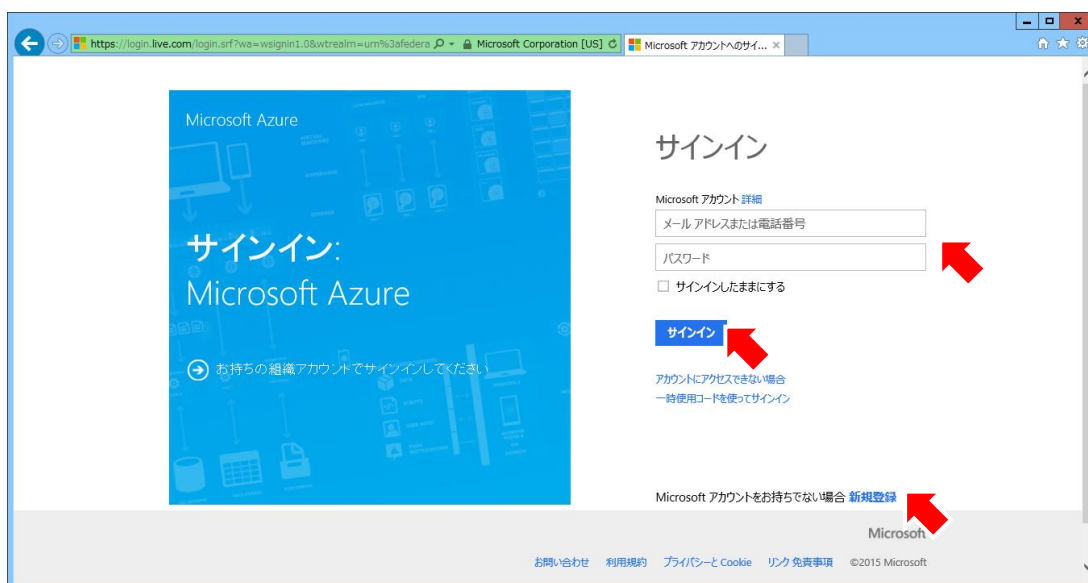
## 2.1 Microsoft Azure 無料評価版のサインアップ手順

この手順はオプションです。Microsoft Azure サブスクリプションをお持ちでない場合、次の手順を実行し、Microsoft Azure の 1 ヶ月間の無料評価版のサブスクリプションを取得してください。

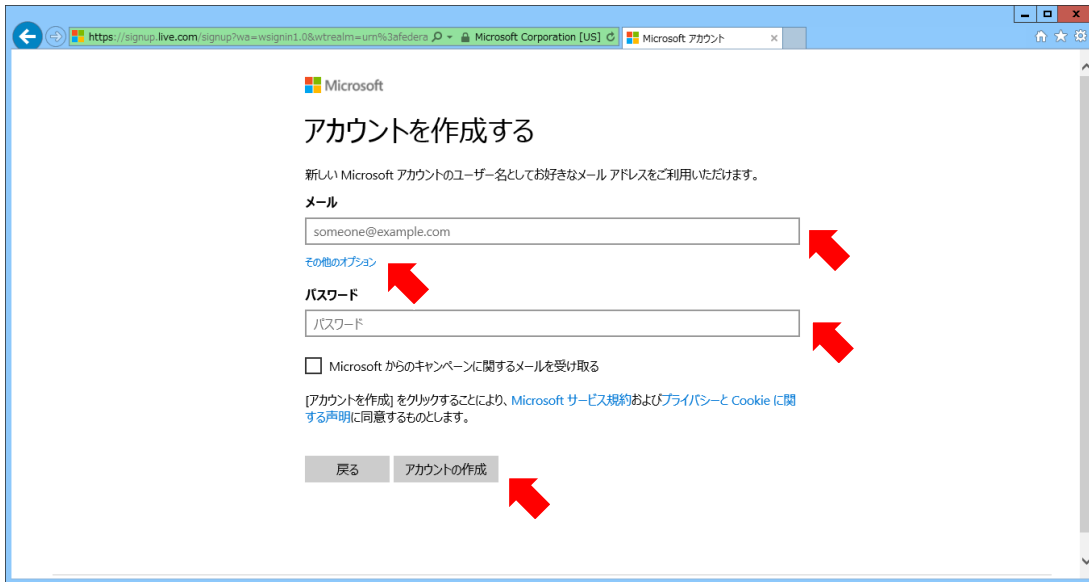
1. Web ブラウザーを起動し、[1 か月間の無料試用版](#) のページにアクセスします。
2. [1 か月間の無料評価版] が表示されます。[今すぐ試す] をクリックします。



3. [サインイン] が表示されます。Microsoft アカウントのメール アドレスとパスワードを入力し、[サインイン] ボタンをクリックします。また、Microsoft アカウントをお持ちでない場合は、[新規登録] をクリックします。



4. [新規登録] をクリックした場合、[アカウントを作成する] が表示されます。自分のメールアドレス（自分が日常的に使用しているメールアドレス）と任意のパスワードを指定し、[アカウントの作成] ボタンをクリックします。なお、利用できるメールアドレスを持ちでない場合、[その他のオプション] をクリックし、新しい Outlook メール（xxxxxxx@outlook.jp）を作成することもできます。



#### ワンポイント

ここで登録した Microsoft アカウントは Microsoft Azure サブスクリプションの管理者（サブスクリプション所有者）になります。

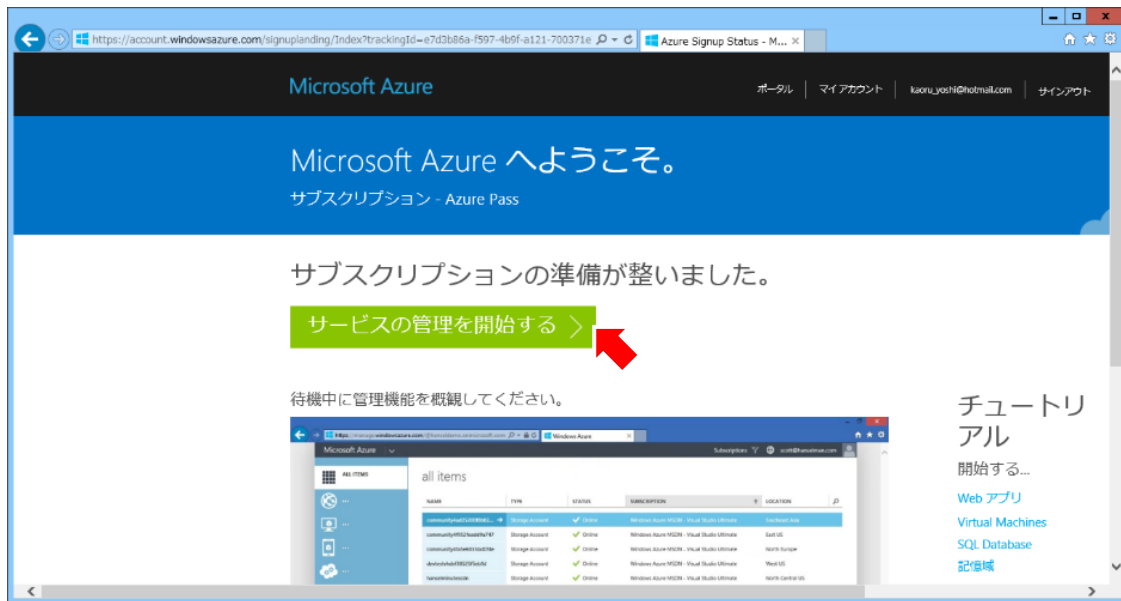
5. [サインアップ] が表示されます。サインアップに必要な情報を指定します。なお、サインアップには、ショートメッセージ（SMS）またはコールバックを利用した電話による確認とクレジットカードの登録による確認が必要です。確認が完了したら、[サインアップ] をクリックします。



#### ワンポイント

ここで登録したクレジットカードが勝手に使用されることはありません。

6. [Microsoft Azure へようこそ] の [準備が完了するまでしばらくお待ちください] が表示されます。しばらく待つと、[サブスクリプションの準備が整いました] が表示されます。[サービスの管理を開始する] ボタンをクリックします。



#### ワンポイント

Microsoft Azure の 1 か月間の無料評価版では、20,500 円相当のクレジットが利用可能です。なお、無料評価版の利用は 1 回までとなっており、過去すでに利用された方は無料評価版にサインアップいただけませんので、ご注意ください。なお、無料評価版の使用制限に達した場合の動作については、以下の情報も参考にしてください。

- [お知らせ] 無料評価版の使用制限に達した場合の動作について

<http://blogs.msdn.com/b/dsazurejp/archive/2012/12/28/windows-azure-90-days-free-offering.aspx>

引き続き、ご使用される場合は、有償のサブスクリプションに切り替えていただきますようお願いいたします。また、既に MSDN Subscription をお持ちの方やスタートアップ企業の方、大学などの教育機関でご利用の方は、以下より詳細をご確認ください。

- MSDN サブスクライバー向けの Azure の特典

<https://azure.microsoft.com/ja-jp/pricing/member-offers/msdn-benefits-details/>

- スタートアップのための BizSpark

<https://azure.microsoft.com/ja-jp/pricing/member-offers/bizspark-startups/>

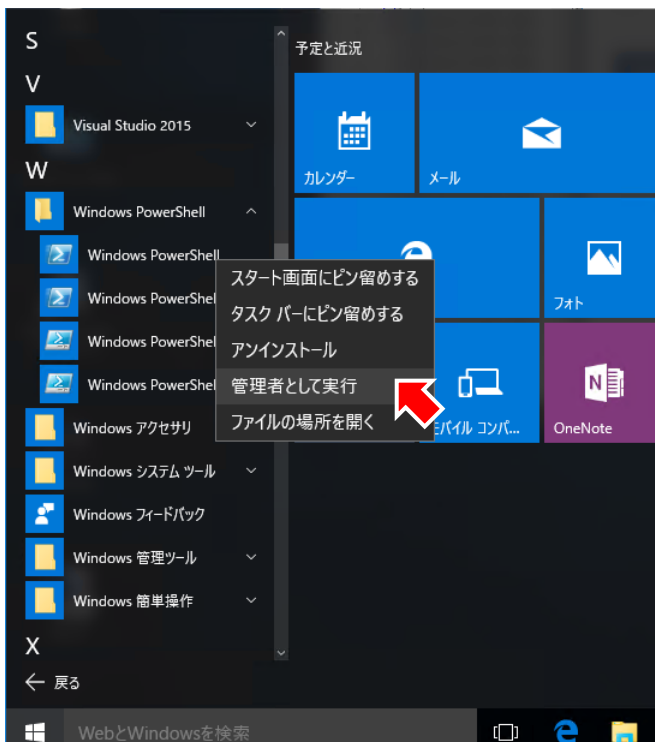
- 教育機関でのご利用

<https://www.microsoftazurepass.com/azureu>

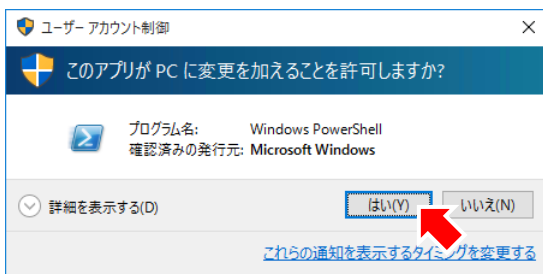
## 2.2 Azure PowerShell 1.0.1 モジュールのインストール手順

ここでは、Windows 10 を実行しているコンピューターで Azure PowerShell 1.0.1 モジュールをインストールして、AzureResourceManager モードを使用する手順を説明します。なお、Windows 10 以前の OS (Windows Server 2012 R2、Windows 8.1 Pro、Windows 8.1 Enterprise、Windows Server 2012、Windows 7、Windows Server 2008 R2) が稼働するコンピューターで実習を行う場合、Azure PowerShell 1.0 1 モジュールのインストール前に、Windows Management Framework 5.0 (<http://www.microsoft.com/en-us/download/details.aspx?id=48729>) をインストールしておく必要があります。

1. Windows のスタート メニューの [すべてのアプリケーション] から [Windows PowerShell] を右クリックして、[管理者として実行] を選択し、Windows PowerShell を管理者として起動します。

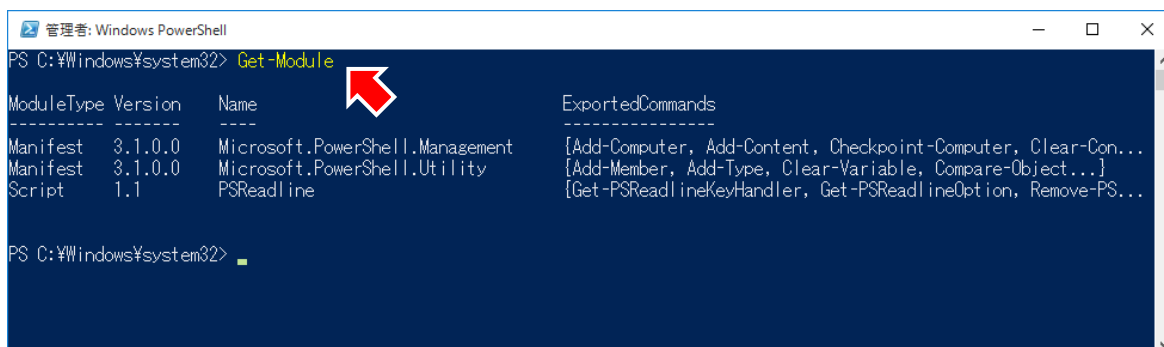


2. [ユーザー アカウント制御] が表示されたら [はい] ボタンをクリックします。



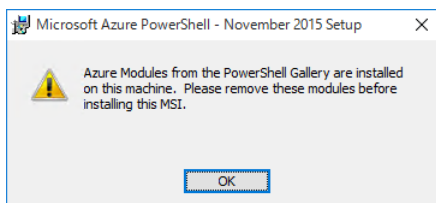
3. Windows PowerShell ウィンドウに次のコマンドを入力し、Enter キーを押して実行し、インポートされている PowerShell モジュールを確認します。

Get-Module

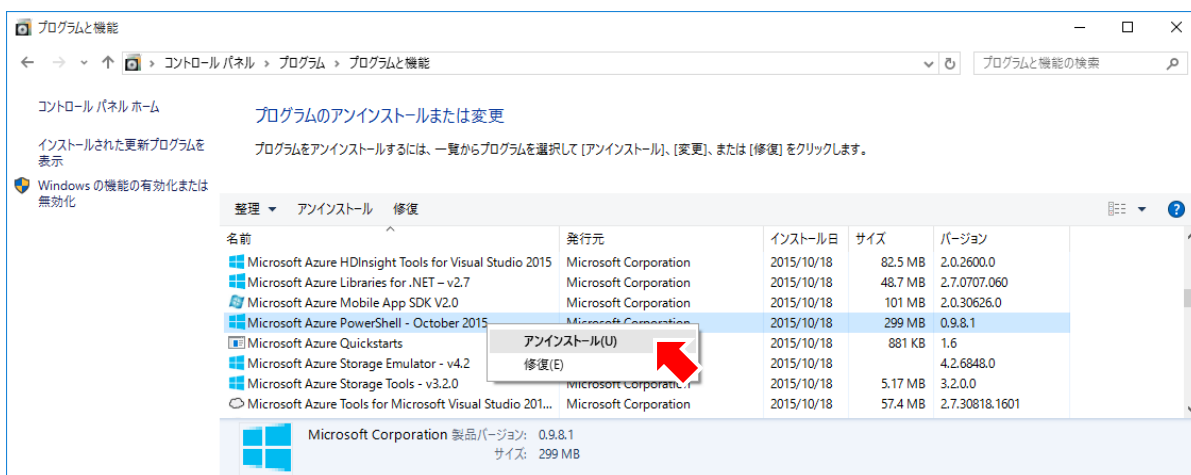


### ワンポイント

Get-Module コマンドレットは、現在、インポートされているモジュールの情報を取得します。このコマンドの出力に、Name 列が「Azure」で Version 列が「0.9.8.1」以前の Azure PowerShell モジュールが表示された場合、以降に実行するインストールプログラムの実行で、次のようなメッセージが表示されます。



この場合、事前に、コントロールパネルから古い Azure PowerShell を削除しておく必要があります。

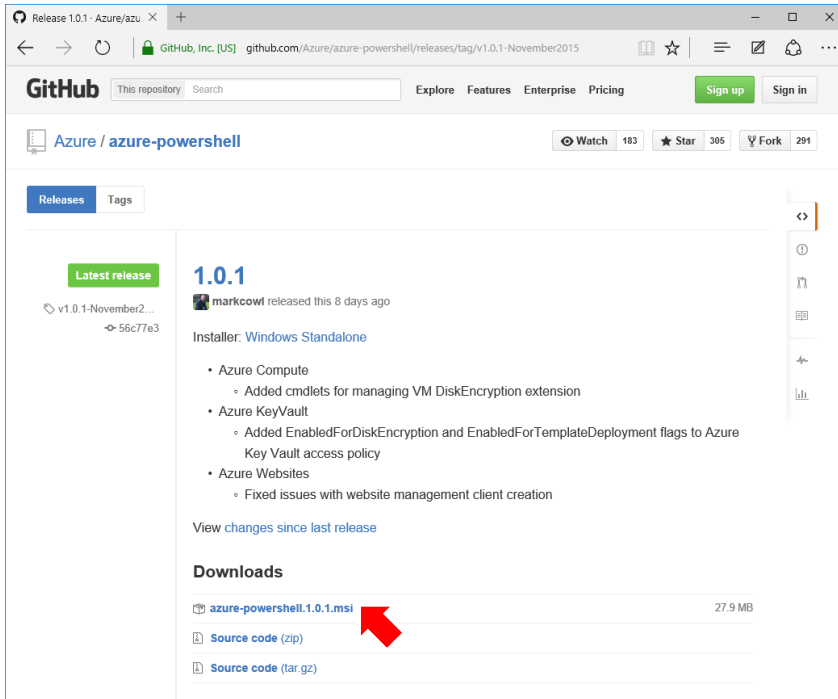


また、過去に、「1.0.0」のモジュールをインストールしている場合は、事前に以下の PowerShell コマンドを実行して、古いモジュールを削除した後、以降の手順を実行してください。

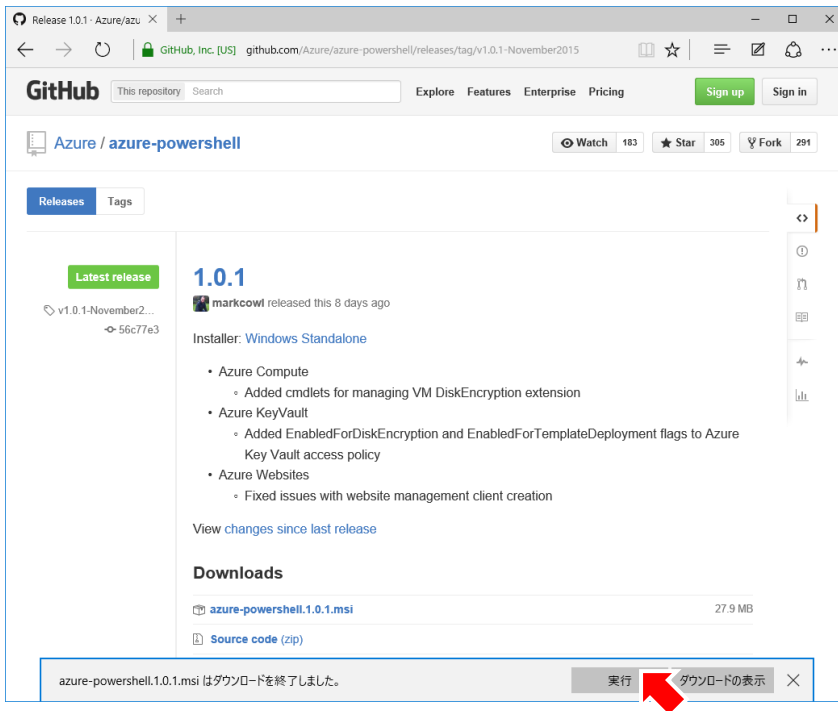
```
Uninstall-AzureRM
Uninstall-Module AzureRM
```



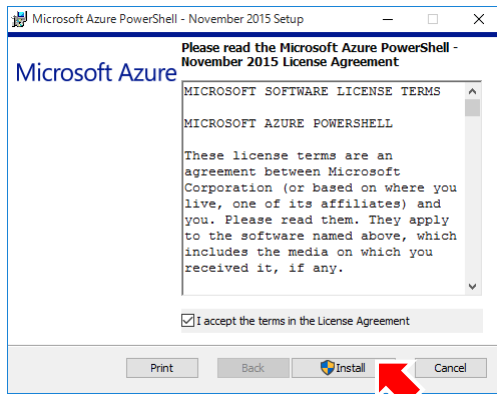
- Web ブラウザーで GitHub の「Azure / azure-powershell」の 1.0.1 のページ (<https://github.com/Azure/azure-powershell/releases/tag/v1.0.1-November2015>) にアクセスします。[Downloads] の [azure-powershell.1.0.1.msi] をクリックして、インストール モジュールのダウンロードを開始します。



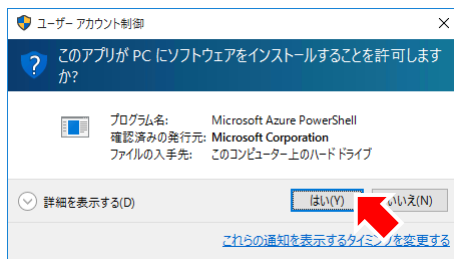
- ダウンロードが完了すると、Web ブラウザーの下に、ボタンが表示されるので、[実行] ボタンをクリックします。



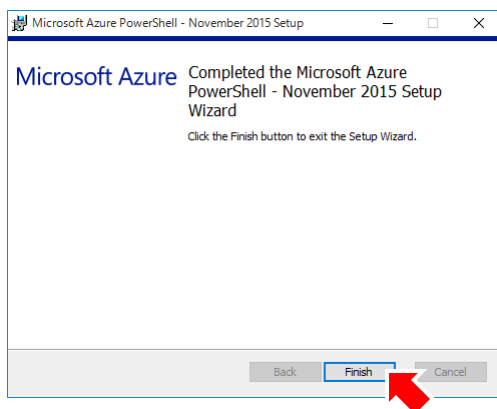
6. [Microsoft Azure PowerShell –November 2015 Setup] が起動するので [Install] ボタンをクリックします。



7. [ユーザー アカウント制御] が表示されたら、[はい] ボタンをクリックします。

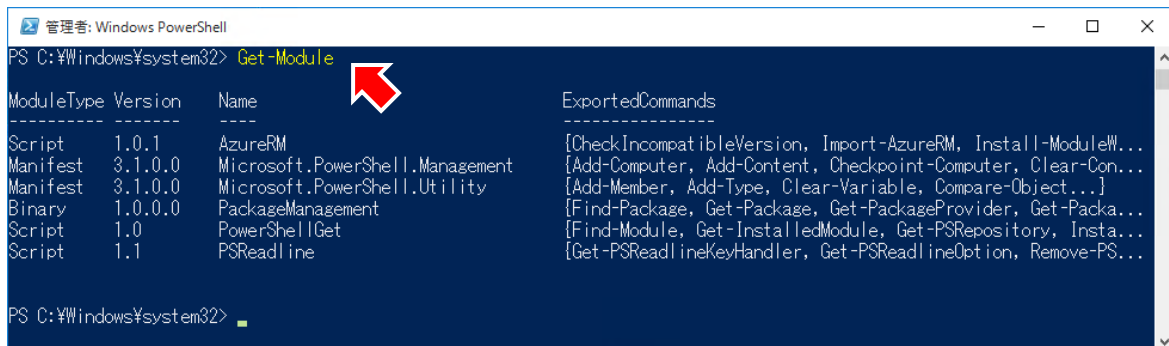


8. インストールが完了したら [Finish] ボタンをクリックして、終了します。



9. Windows PowerShell ウィンドウに次のコマンドを入力し、Enter キーを押して実行し、AzureRM 1.0.1 のモジュールがインストールされていることを確認します。

```
Get-Module
```

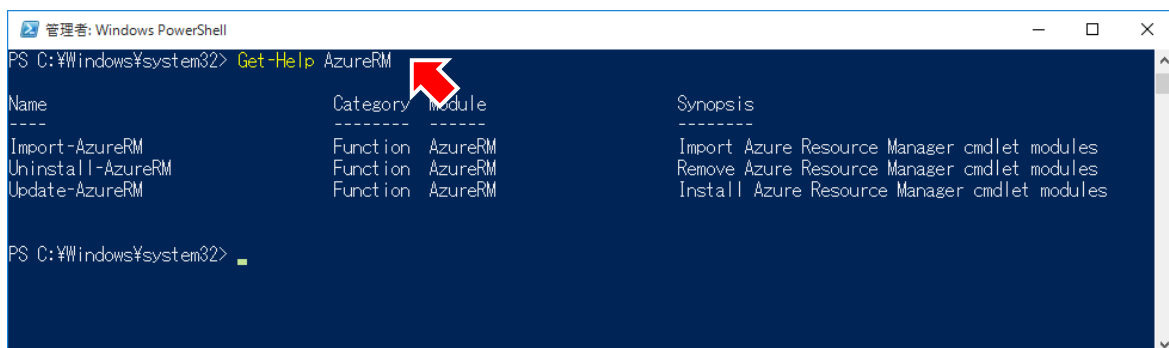


```
管理: Windows PowerShell
PS C:\Windows\system32> Get-Module

ModuleType Version Name ExportedCommands
-----
Script 1.0.1 AzureRM [CheckIncompatibleVersion, Import-AzureRM, Install-ModuleW...
Manifest 3.1.0.0 Microsoft.PowerShell.Management [Add-Computer, Add-Content, Checkpoint-Computer, Clear-Con...
Manifest 3.1.0.0 Microsoft.PowerShell.Utility [Add-Member, Add-Type, Clear-Variable, Compare-Object...
Binary 1.0.0.0 PackageManagement [Find-Package, Get-Package, Get-PackageProvider, Get-Packa...
Script 1.0 PowerShellGet [Find-Module, Get-InstalledModule, Get-PSRepository, Insta...
Script 1.1 PSReadline [Get-PSReadlineKeyHandler, Get-PSReadlineOption, Remove-PS...
```

10. Windows PowerShell ウィンドウに次のコマンドを入力し、Enter キーを押して実行し、モジュールのヘルプ情報を取得します。

```
Get-Help AzureRM
```

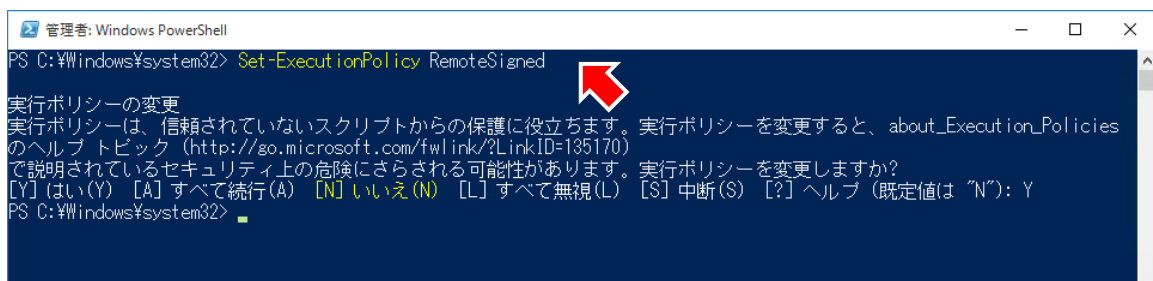


```
管理: Windows PowerShell
PS C:\Windows\system32> Get-Help AzureRM

Name Category Module Synopsis
----
Import-AzureRM Function AzureRM Import Azure Resource Manager cmdlet modules
Uninstall-AzureRM Function AzureRM Remove Azure Resource Manager cmdlet modules
Update-AzureRM Function AzureRM Install Azure Resource Manager cmdlet modules
```

11. スクリプトを実行する場合、以下のコマンドを入力し、選択状態にして、[選択項目を実行] ボタンで実行し、現在の実行ポリシーを RemoteSigned に変更します。メッセージが表示されたら、[Y] を Enter します。

```
Set-ExecutionPolicy RemoteSigned
```



```
管理: Windows PowerShell
PS C:\Windows\system32> Set-ExecutionPolicy RemoteSigned

実行ポリシーの変更
実行ポリシーは、信頼されていないスクリプトからの保護に役立ちます。実行ポリシーを変更すると、about_Execution_Policies
のヘルプ トピック (http://go.microsoft.com/fwlink/?LinkID=135170)
で説明されているセキュリティ上の危険にさらされる可能性があります。実行ポリシーを変更しますか?
[Y] はい(Y) [A] すべて続行(A) [N] いいえ(N) [L] すべて無視(L) [S] 中断(S) [?] ヘルプ (既定値は "N"): Y
PS C:\Windows\system32>
```

## 2.3 認証とサブスクリプションの選択

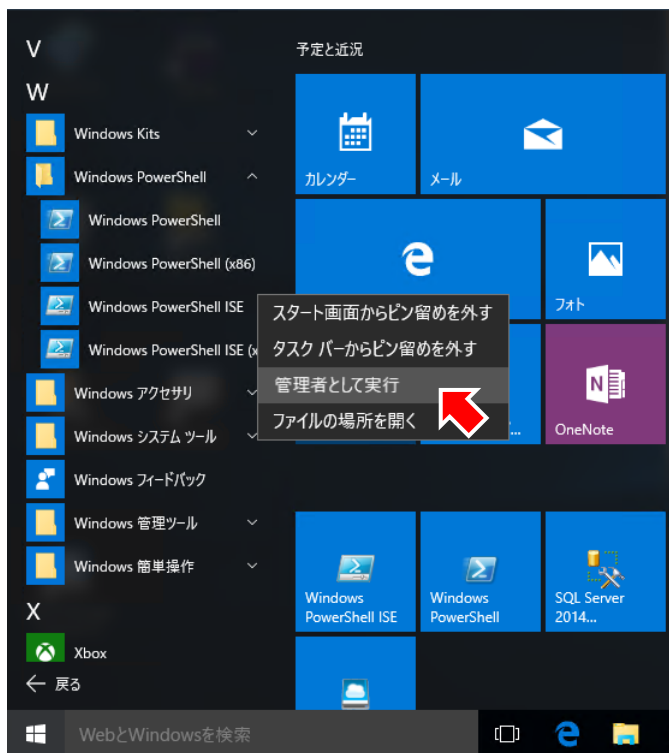
Azure PowerShell コマンドレットで Microsoft Azure に接続し、操作するには、資格情報を使用した認証が必要です。資格情報を認証するには次の 2 種類の方法が選択できます。

- Azure Active Directory (Azure AD) による認証
- 管理証明書を使用する認証

ARM モードのコマンドレットでは、現在、Azure AD による認証のみがサポートされ、Azure ポータルから発行設定ファイルをダウンロードして行う管理証明書を使用する認証は使用できません。ここでは、Azure AD による認証方式を説明します。この方式では、Login-AzureRmAccount コマンドレットにより、対話形式で認証を行い Azure Resource Manager に接続します。

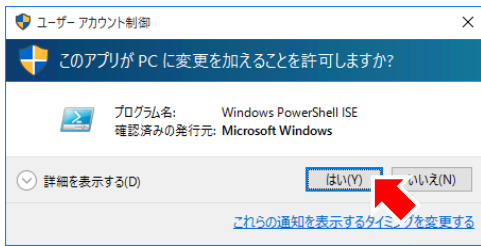
ここからのスクリプティング作業は、Windows PowerShell ISE を使用して行います。

1. Windows のスタート メニューの [すべてのアプリケーション] から [Windows PowerShell ISE] を右クリックして、[管理者として実行] を選択し、Windows PowerShell ISE を管理者として起動します。

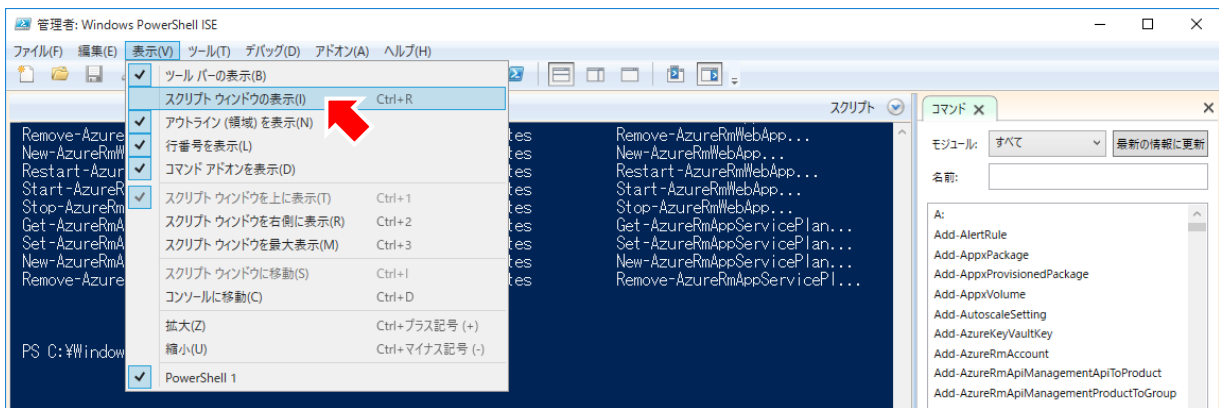


## Microsoft Azure 自習書シリーズ コマンドラインによる Azure の管理

2. [ユーザー アカウント制御] が表示されたら [はい] ボタンをクリックします。

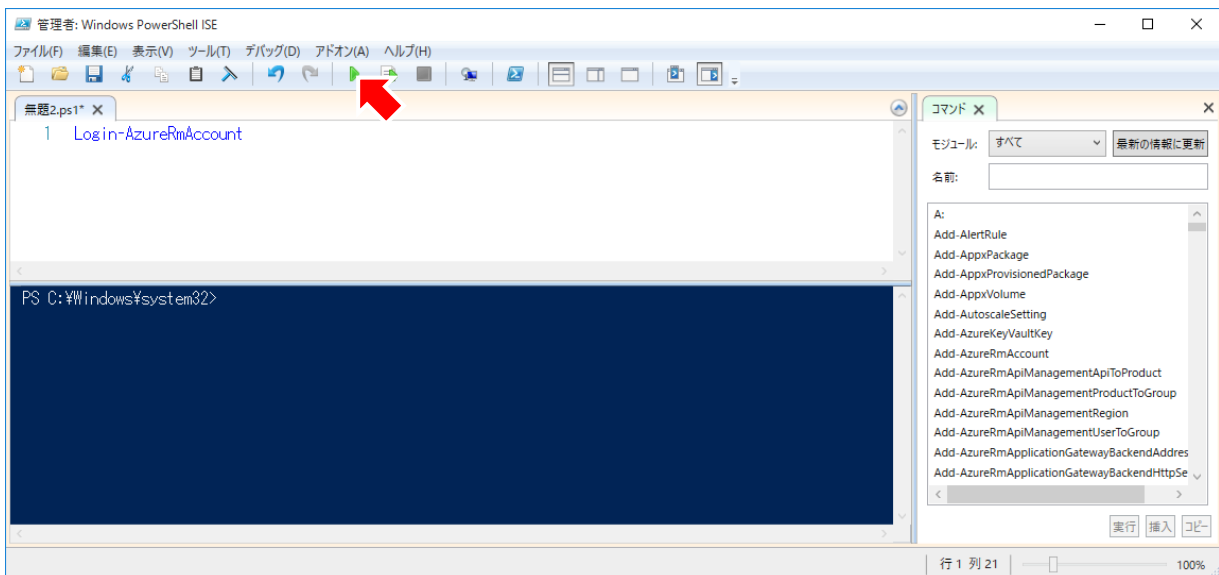


3. [表示] メニューの [スクリプト ウィンドウの表示] を選択して、新しいスクリプト ウィンドウを開きます。

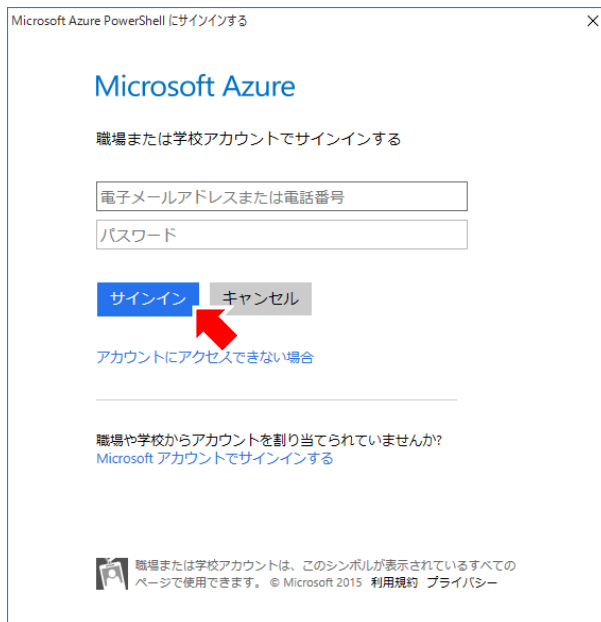


4. スクリプト ウィンドウに以下のコマンドを入力し、[スクリプトを実行] ボタンをクリックして、Microsoft Azure にログインして、Azure Resource Manager に接続します。

Login-AzureRmAccount

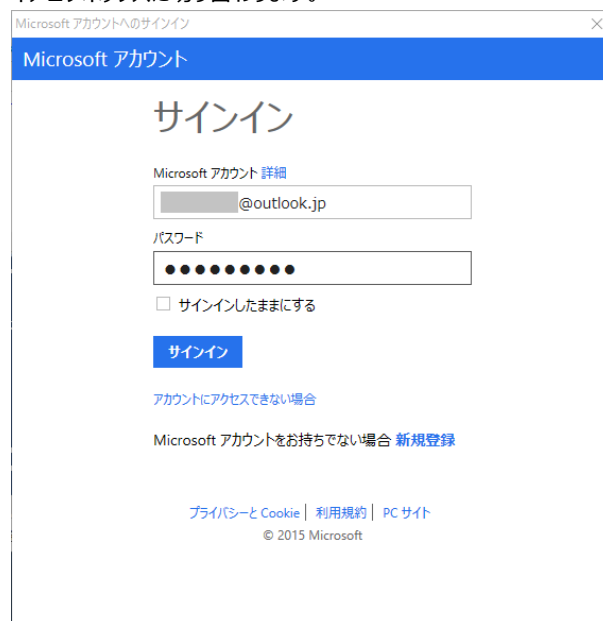


5. 認証画面が表示されるので、使用するアカウントとパスワードを入力して認証します。

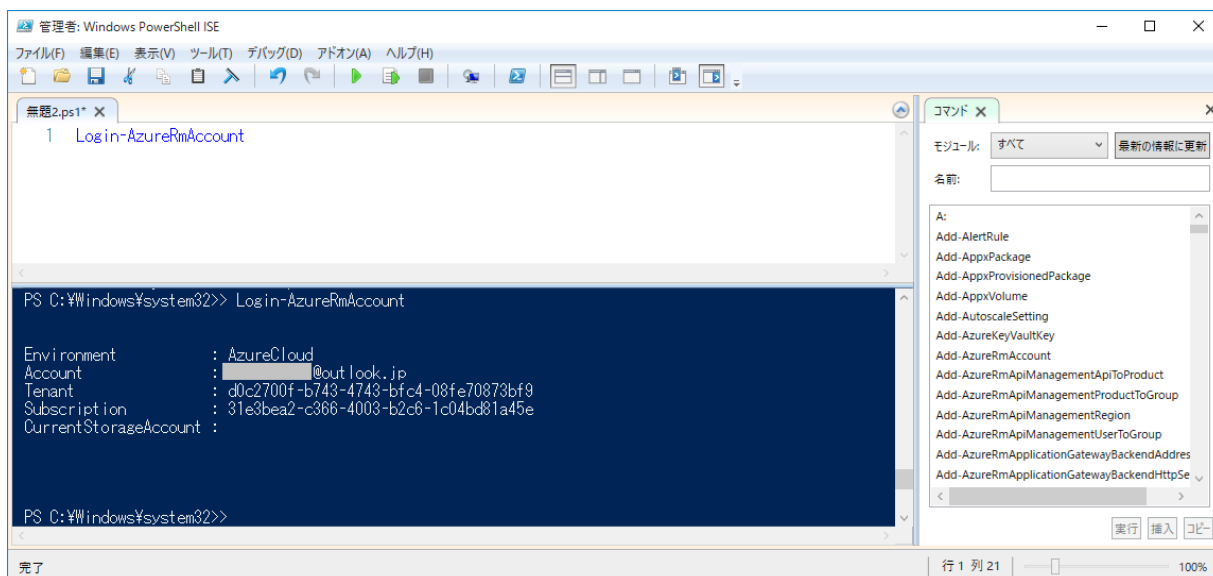


## ワンポイント

上記のダイアログボックスは、Microsoft アカウントを使用して認証する場合、自動的に、次の Microsoft アカウント認証用ダイアログボックスに切り替わります。



6. 正常にログインが完了すると、次のような表示が返されます。



**ワンポイント**

ログインに使用したアカウント情報、AAD のテナント、既定のサブスクリプション、およびクライアントのストレージ アカウントの情報が表示されます。テナントとサブスクリプションの ID は、32 桁の GUID 値として表示されます。

ここで紹介した方法を使用する場合、認証用のダイアログボックスが表示されるため、対話形式での処理が必要となりますが、以下のサンプル コードに示すように Get-Credential コマンドレットで、-Credential を使用して資格情報を渡すことで、ダイアログボックスを表示しないで、認証を行うことも可能です。

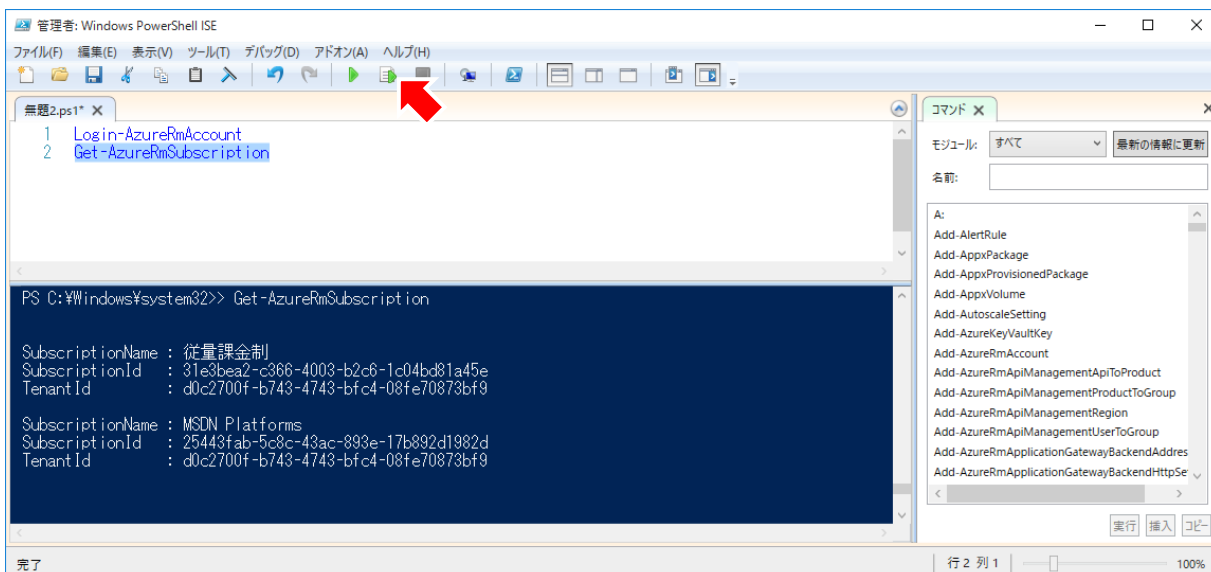
```

$User = "awuser@aw.onmicrosoft.com"
$PassString = "awPa$$w0rd1"
$Pass = ConvertTo-SecureString -String $PassString -AsPlainText -Force
$Cred = New-Object System.Management.Automation.PsCredential($User, $Pass)
Get-Credential -Credential $Cred
Login-AzureRmAccount -Credential $Credential -Verbose
    
```

ただし、この方法は、組織アカウントのみがサポートされ、Microsoft アカウントでは使用することができないことに注意してください。

7. 以下のコマンドを入力し、選択状態にして、[選択項目を実行] ボタンで実行し、アカウントに関連付けられているサブスクリプションを確認します。

```
Get-AzureRmSubscription
```

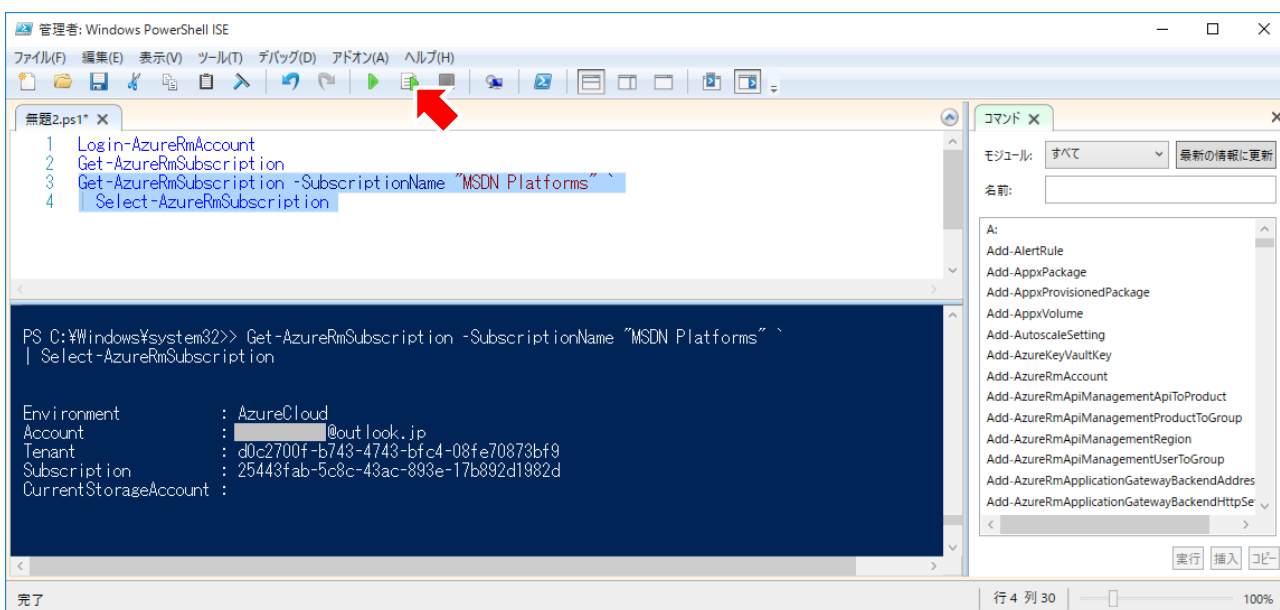


#### ワンポイント

使用しているアカウントに複数のサブスクリプションが関連付けられている場合、Select-AzureRmSubscription コマンドレットを使用して、サブスクリプションを切り替えることができます。

8. アカウントに複数のサブスクリプションが関連付けられている場合、スクリプト ウィンドウに以下のコマンドを入力し、選択状態にして、[選択項目を実行] ボタンで実行し、使用するサブスクリプションを選択します。

```
Get-AzureRmSubscription -SubscriptionName "MSDN Platforms" `
| Select-AzureRmSubscription
```





#### ワンポイント

このコードは、「MSDN Platforms」を使用する例です。-SubscriptionName 引数には、ご自身が使用するサブスクリプション名を入力してください。なお、「MSDN Platforms」の後に記述している「`」は、行継続文字です。これにより 3 行目と 4 行目は、1 行のコードとして扱われます。このドキュメントでは、必要に応じて、長すぎる行は読みやすくするため、行末に「`」（行継続文字）を記述して改行しています。

1. パラメーターとして渡した文字列を変数化するため 先頭に以下の行を追加します。

```
$SubscriptionName = "MSDN Platforms"
```

2. 4 行目を以下に変更します。

```
Get-AzureRmSubscription -SubscriptionName $SubscriptionName `
| Select-AzureRmSubscription
```

3. Windows PowerShell ISE の [ファイル] メニューから [名前を付けて保存] を選択し、ここまで作成したスクリプトを作業用フォルダーに、「**Login-AzureRM.ps1**」という名前で保存します。作成したスクリプトは、以下のコードになります。

```
$SubscriptionName = "MSDN Platforms"
Login-AzureRmAccount
Get-AzureRmSubscription
Get-AzureRmSubscription -SubscriptionName $SubscriptionName `
| Select-AzureRmSubscription
```

4. [Login-AzureRM.ps1] タブを閉じます。

#### ワンポイント

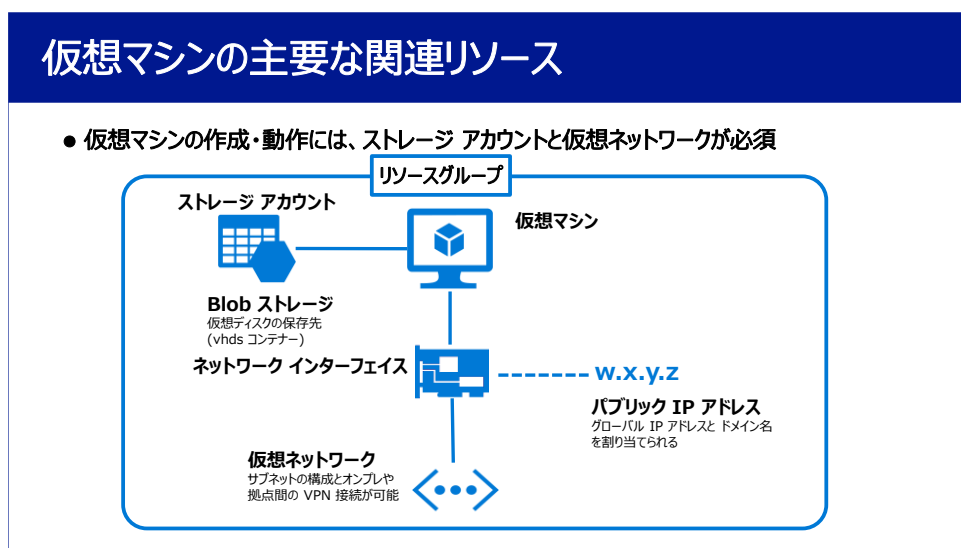
パラメーターを変数化して、スクリプトの先頭部分に宣言しておく可読性が高くなり、再利用性も高まります。保存したファイルは、\$subscriptionName 変数にご自身が使用するサブスクリプション名を記述した後、[スクリプトを実行] ボタンをクリックするか、保存したファイルを右クリックして [PowerShell で実行] を選択することで、スクリプトとしてまとめて実行することができます。

なお、ここで紹介した認証方法は、Azure PowerShell コマンドレットで Azure Resource Manager (ARM) に接続していることに注意してください。従来の Azure Service Management (ASM) に接続する必要がある場合、これまで通り「Install-Module Azure」で ASM のモジュールをダウンロードして、「Add-AzureAccount」コマンドレットを実行して ASM に接続して認証する必要があります。

### 3. PowerShell による IaaS v2 リソースの構築

仮想マシンの作成では、ストレージ アカウントと仮想ネットワークは必須のリソースです。Azure ストレージ アカウントを使用すると、Azure Blob サービス、キュー サービス、テーブル サービス、ファイル サービスにアクセスできます。

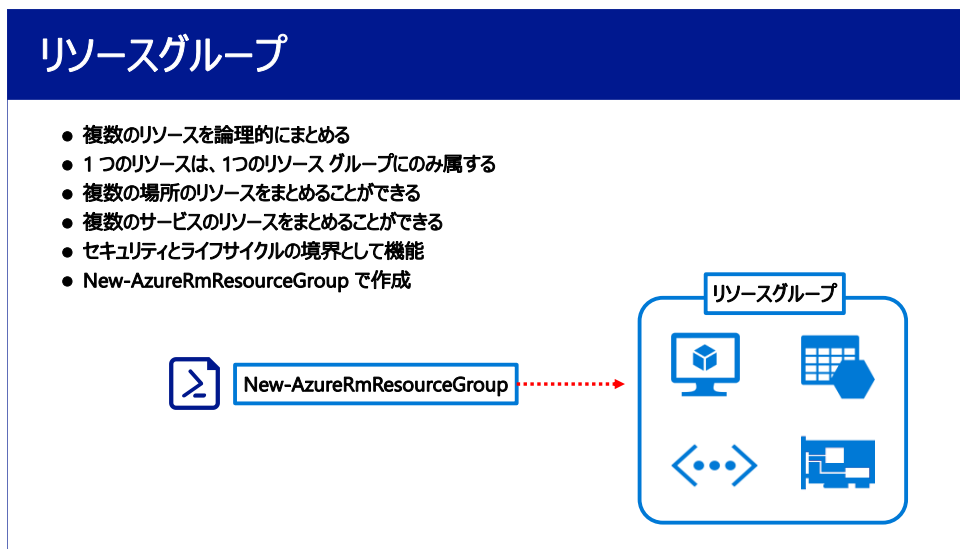
仮想マシンの VHD ファイルは、既定で Blob サービスの vhds コンテナに保存されます。仮想マシンは、ネットワーク インターフェイス リソースを介して仮想ネットワークに参加します。ネットワーク インターフェイス リソースには、グローバル インターネットに接続するためのパブリック IP アドレスと仮想ネットワーク内で通信するためのプライベート IP アドレスを割り当てることができます。パブリック IP アドレスは、パブリック IP アドレス リソースを作成して割り当てます。



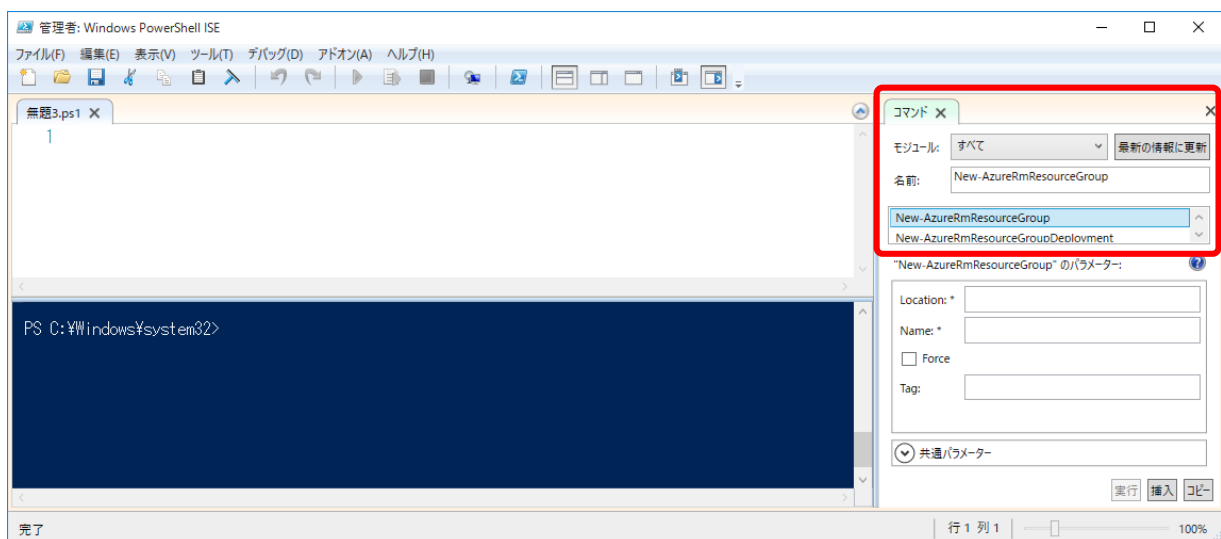
本章では、これらの仮想マシンの主要な関連リソースを PowerShell コマンドレットを使用して作成します。なお、「3.1 リソース グループの作成」から「3.3 仮想ネットワークとサブネットの作成」までは、Windows PowerShell ISE のコマンド アドオン ウィンドウの使用方法に慣れてもらうため、この機能を使用して、スクリプトを作成する手順を紹介します。「3.4 ネットワーク インターフェイスとパブリック IP アドレスの作成」以降は、作成するスクリプトを中心に説明しますが、コマンドレットに必要なパラメーターを確認するような場合は適宜、コマンド アドオン ウィンドウを使用していただけるとよいでしょう。

### 3.1 リソース グループの作成

リソース グループは、複数のリソースを論理的にまとめることができるグループです。リソースグループは、展開、更新、削除に至るリソースのライフサイクル管理の境界として機能します。また、サービスの管理者は、Azure AD に登録されたユーザーやグループに対し、リソース グループ、またはリソース単位で役割ベースのアクセス制御を構成することができます。最初に、New-AzureRmResourceGroup コマンドレットでリソース グループを作成します。



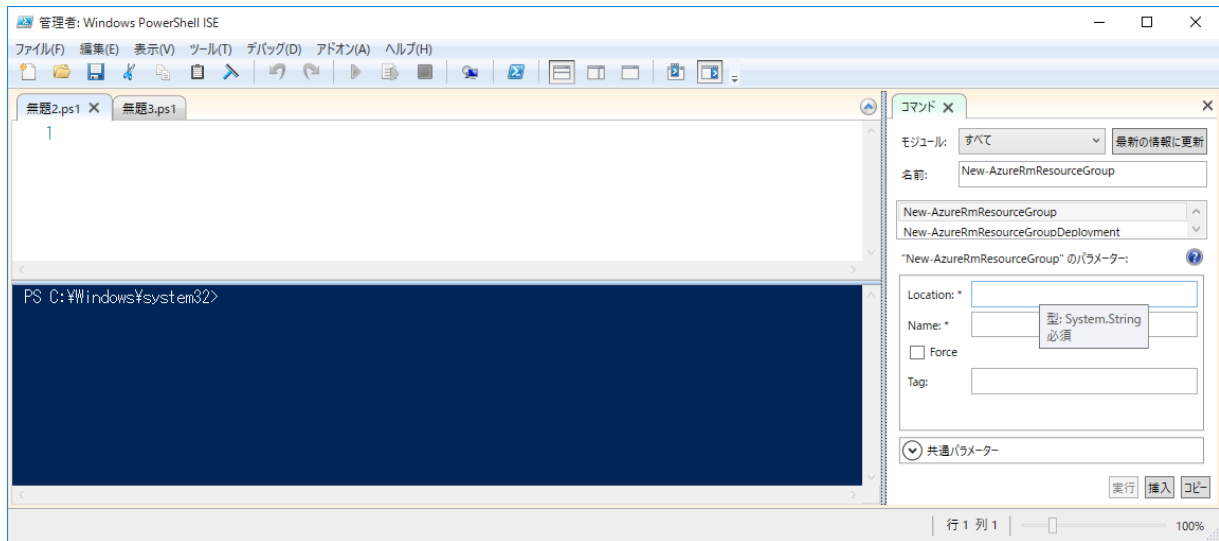
1. Windows PowerShell ISE の [表示] メニューの [スクリプト ウィンドウの表示] を選択して、新しいスクリプト ウィンドウを開きます。
2. New-AzureRmResourceGroup コマンドレットの引数を確認するため、コマンド アドオン ウィンドウの [名前] に「New-AzureRmResourceGroup」と入力し、[詳細の表示] ボタンが表示されたら、ボタンをクリックして、リストボックスに表示される「New-AzureRmResourceGroup」を選択します。



#### ワンポイント

この操作で、[New-AzureRmResourceGroup のパラメーター] が表示され、必須のパラメーターとして、「Location」と「Name」を渡す必要があることを確認します。必須のパラメーターには、「\*」が表示されます。

3. コマンド アドオン ウィンドウの [New-AzureRmResourceGroup のパラメーター] の下に表示されている [Location] テキストボックスの上にマウスを移動します。

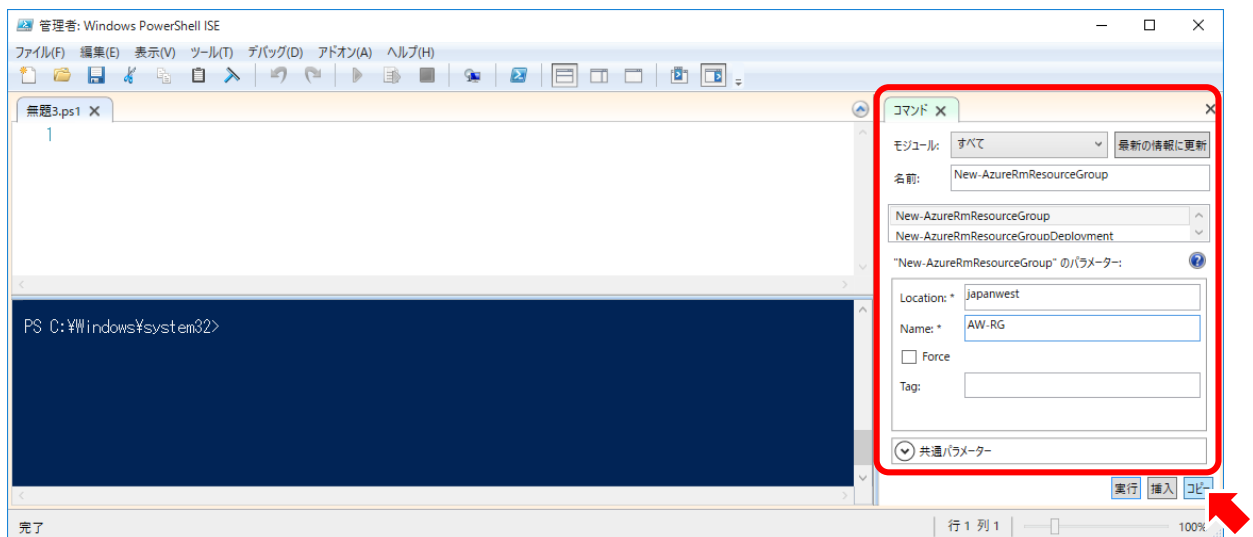


**ワンポイント**

この操作で、[Location] に設定できる値は文字列 (System.String) で、必須のパラメーターであることがわかります。必須パラメーターには、パラメーター名の右上に「\*」が表示されています。New-AzureRmResourceGroup には、パラメーターとして、「Location」と「Name」を必ず渡す必要があることを意味します。

4. コマンド アドオン ウィンドウの [New-AzureRmResourceGroup のパラメーター] に下表の設定値を入力して、[コピー] ボタンをクリックします。

パラメーター	説明	設定値
<b>Location</b>	仮想マシンを作成する地域を入力します。	japanwest
<b>Name</b>	リソース グループ名を入力します。	AW-RG

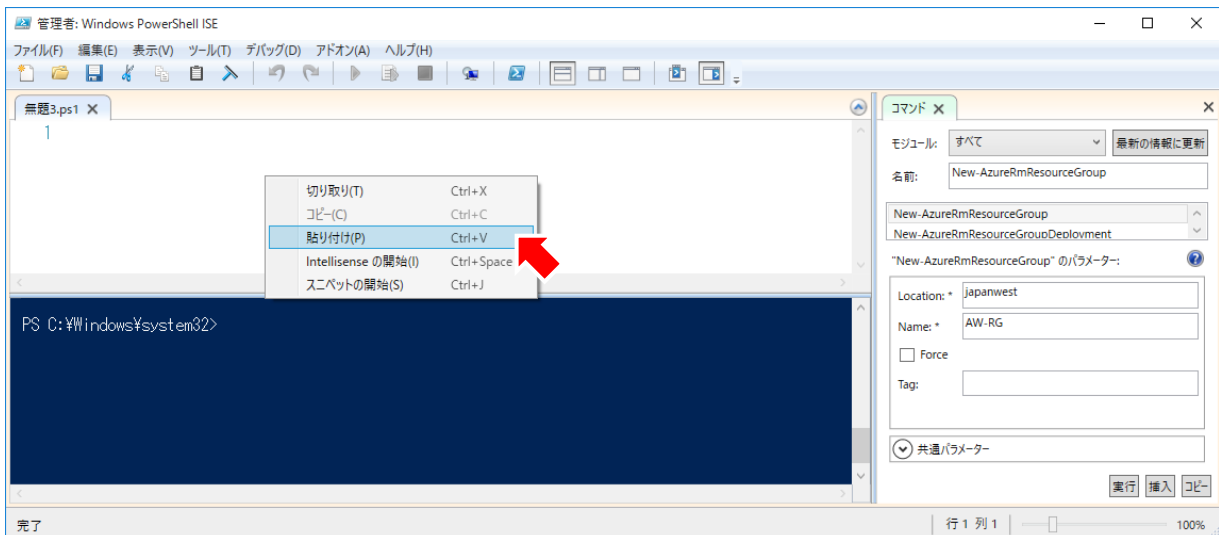


### ワンポイント

New-AzureRmResourceGroup の Location の値として設定できる地域名は、下表を参照してください。

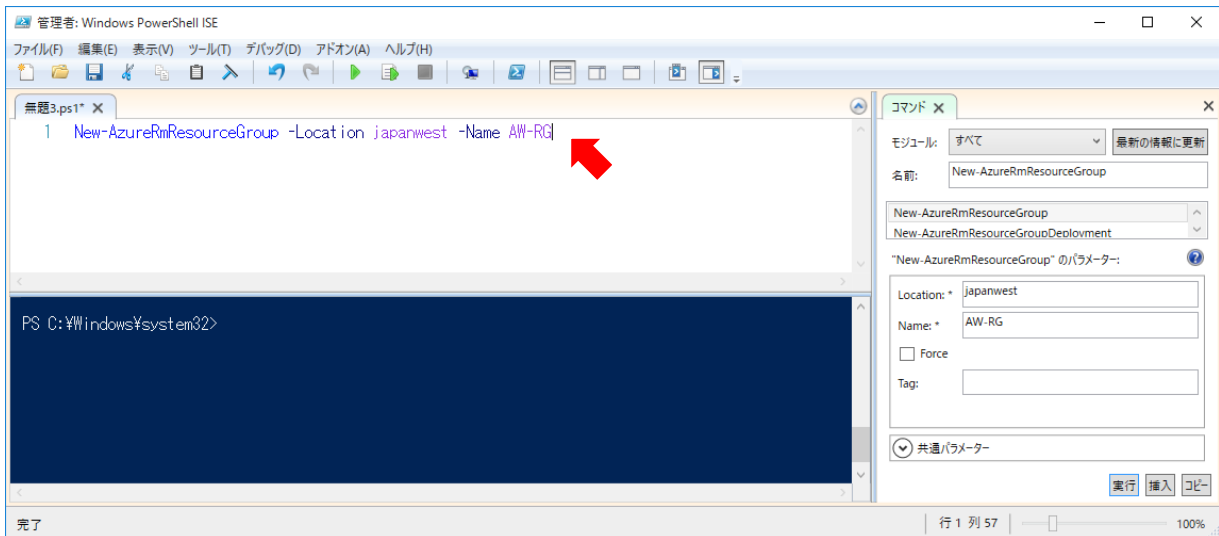
地域	Location の値
東アジア	eastasia
東南アジア	southeastasia
西日本	japanwest
東日本	japaneast
米国西部	westus
米国中央部	centralus
米国中南部	southcentralus
米国中北部	northcentralus
米国東部	eastus
米国東部 2	eastus2
ブラジル南部	brazilsouth
ヨーロッパ西部	westeurope
ヨーロッパ北部	northeurope

5. スクリプト ウィンドウの任意の箇所を右クリックして、[貼り付け] を選択します。

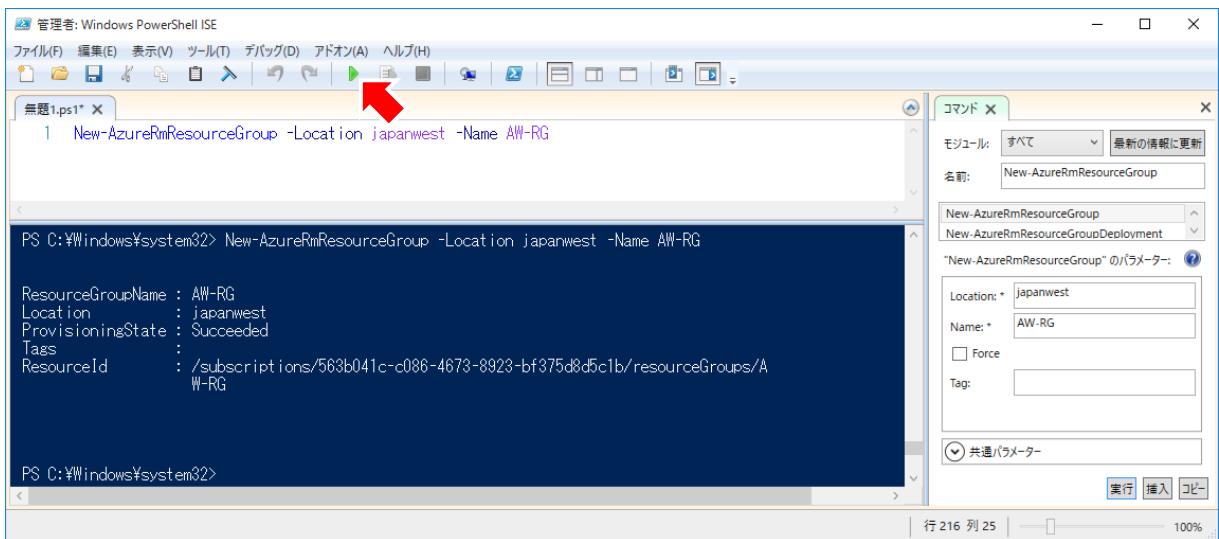


6. 以下のコマンドが入力されたことを確認します。

```
New-AzureRmResourceGroup -Location japanwest -Name AW-RG
```

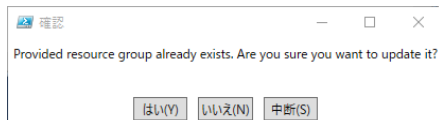


7. [スクリプトを実行] ボタンをクリックして実行し、リソース グループを作成します。



### ワンポイント

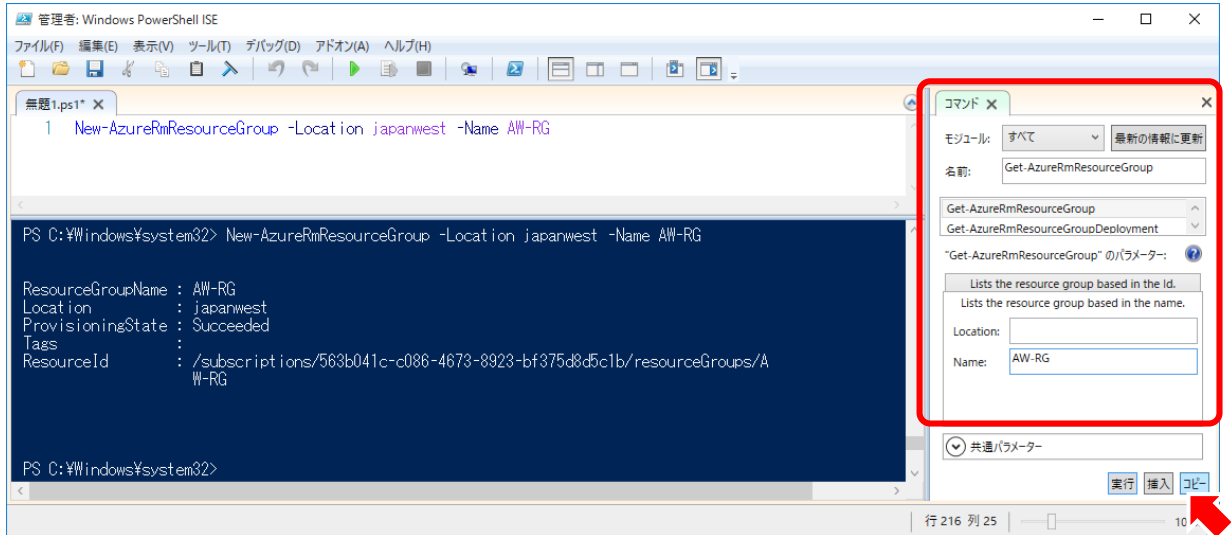
このコマンドレットの実行により返される結果で、ProvisioningStatus が Succeeded になることを確認します。なお、以下のメッセージ ボックスが表示される場合は、すでに、「AW-RG」リソースグループが作成されています。この場合は、[いいえ] ボタンをクリックして、まず、既存の「AW-RG」リソースグループを削除してから、実習を行うか、または、リソースグループ名を「AW-RG」からほかの名前に変更して実習を行ってください。



その場合は、以降の手順で、「AW-RG」の記載があった場合、すべて、ご自身が作成したリソースグループ名を使用するように名前を置き換えてください。

8. コマンド アドオン ウィンドウの [名前] に「**Get-AzureRmResourceGroup**」と入力し、リストボックスに表示される「Get-AzureRmResourceGroup」を選択します。コマンド アドオン ウィンドウの [Get-AzureRmResourceGroup のパラメーター] に下表の設定値を入力して、[コピー] ボタンをクリックします。

パラメーター	説明	設定値
Name	リソース グループ名を入力します。	AW-RG

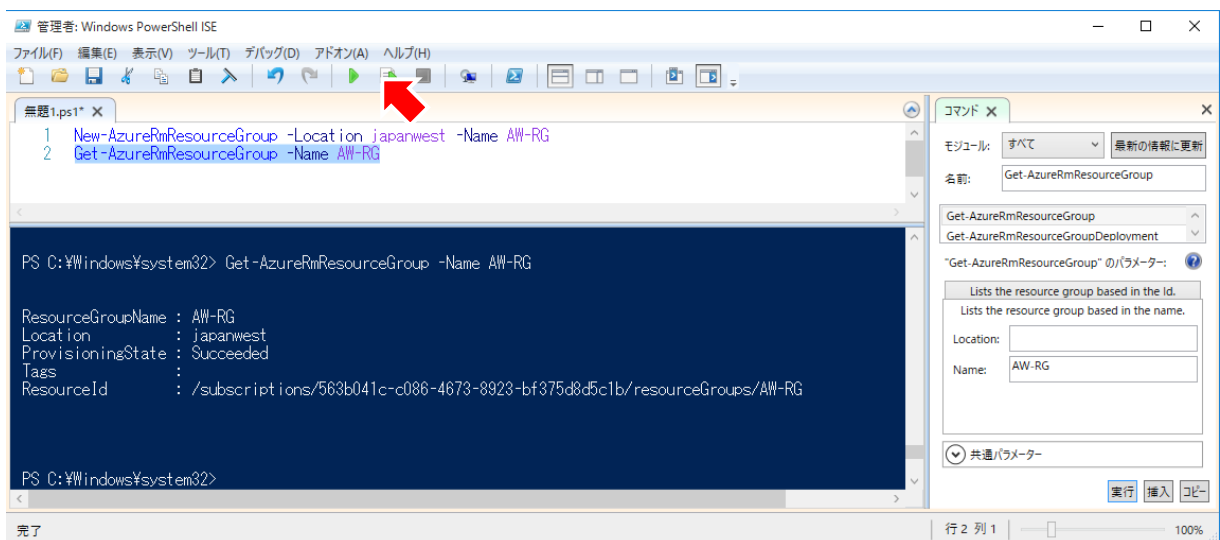


#### ワンポイント

この操作で、[Get-AzureRmResourceGroup のパラメーター] が表示され、任意のパラメーターとして、「Location」と「Name」を指定できることを確認します。これらのパラメーターは省略することができます。省略した場合、サブスクリプションに作成されたすべてのリソースグループが表示されます。

9. スクリプト ウィンドウの 1 行目の最後で改行して 2 行目で右クリックして、[貼り付け] を選択します。この操作で、以下のコマンドが入力されたことを確認して、選択状態にして、[選択項目を実行] ボタンで実行し、作成したリソースグループの情報を確認します。

```
Get-AzureRmResourceGroup -Name AW-RG
```



#### ワンポイント

「Get-」で始まるコマンドレットは、対象となるオブジェクトのプロパティを取得し、表示することができます。この情報を変数に格納することで、オブジェクトの任意のプロパティの参照や操作が行えます。

10. パラメーターとして渡した文字列を変数化するため、スクリプトウィンドウの先頭行に以下の行を追加します。

```
$rg = "AW-RG"  
$location = "japanwest"
```

11. 記述した編集を使用するように、3 行目と 4 行目を以下に変更します。

```
New-AzureRmResourceGroup -Location $location -Name $rg  
Get-AzureRmResourceGroup -Name $rg
```

12. Windows PowerShell ISE の [ファイル] メニューから [名前を付けて保存] を選択し、ここまで作成したスクリプトを作業用フォルダーに、「**Create-ResourceGroup.ps1**」という名前で保存します。作成したスクリプトは、以下のコードになります。

```
$rg = "AW-RG"  
$location = "japanwest"  
New-AzureRmResourceGroup -Location $location -Name $rg  
Get-AzureRmResourceGroup -Name $rg
```

13. [Create-ResourceGroup.ps1] タブを閉じます。



## 3.2 ストレージ アカウントの作成

ここでは、仮想マシンの VHD ファイルを格納するために使用するストレージ アカウントを作成します。ストレージ アカウントは、New-AzureRmStorageAccount で作成します。

### ストレージ アカウント

- Blob、テーブル、キュー、ファイルにアクセスするための最上位レベルの名前空間
- 仮想マシン ディスク (VHD) ファイルの格納に vhds コンテナのページ Blob を使用
- 冗長化オプションを指定
  - ローカル冗長ストレージ (Standard\_LRS)
  - 地理的冗長ストレージ (Standard\_GRS)
  - 読み取りアクセス地理的冗長ストレージ (Standard\_RAGRS)
  - ローカル冗長プレミアム ストレージ (Premium\_LRS)
- New-AzureRmStorageAccount で作成

New-AzureRmStorageAccount コマンドレットには、下表のパラメータを指定します。

パラメーター	説明	必須
<b>Location</b>	ストレージ アカウントを作成する地域を入力します。	○
<b>Name</b>	ストレージ アカウントの名前を入力します。	○
<b>ResourceGroupName</b>	ストレージ アカウントを作成するリソース グループ名を入力します。	○
<b>Type</b>	ストレージの冗長化オプションを選択します。	○
<b>Tags</b>	ストレージ アカウントに対するタグを設定します。	

### ワンポイント

仮想マシンを作成する場合、ストレージ アカウントの Type には、以下のストレージ冗長化オプションが選択できます。

Type の値	冗長化オプション	説明
<b>Standard_LRS</b>	ローカル冗長ストレージ	データのコピーがデータセンター内に 3 つ保持されます。
<b>Standard_GRS</b>	地理的冗長ストレージ	プライマリ リージョンの 3 つのレプリカに加え、セカンダリ リージョンにも 3 つのレプリカを保管することで、災害復旧対策に対応したサービスの持続性を提供できます。
<b>Standard_RAGRS</b>	読み取りアクセス地理的冗長ストレージ	地理的冗長ストレージの構成で、セカンダリ リージョンのストレージ アカウントに読み取り専用でアクセスすることができます。
<b>Premium_LRS</b>	ローカル冗長プレミアム ストレージ	データのコピーがデータセンター内 3 つ保持される SSD ベースの高速なストレージ

ストレージ アカウントを作成することで、Azure ストレージ サービスに、BLOB、テーブル、キューといった種類の Azure ストレージを作成できるようになります。Azure 仮想マシンの仮想ディスク ファイルの保存には、Blob ストレージを使用します。このストレージには、保存コストに優れた「ブロック Blob」とランダムアクセス性能に優れた「ページ Blob」の 2 種類がありますが、仮想マシンのディスクとして使用されるのは、ページ Blob になります。上の表で示した冗長化オプション以外にも、ゾーン冗長 (Standard\_ZRS) と呼ばれるものがありますが、ブロック Blob でのみ使用可能なため、仮想ディスク ファイルの保存先には選択しません。

1. [表示] メニューの [スクリプト ウィンドウの表示] を選択して、新しいスクリプト ウィンドウを開きます。
2. 最初に以下の文字列変数宣言を記述します。

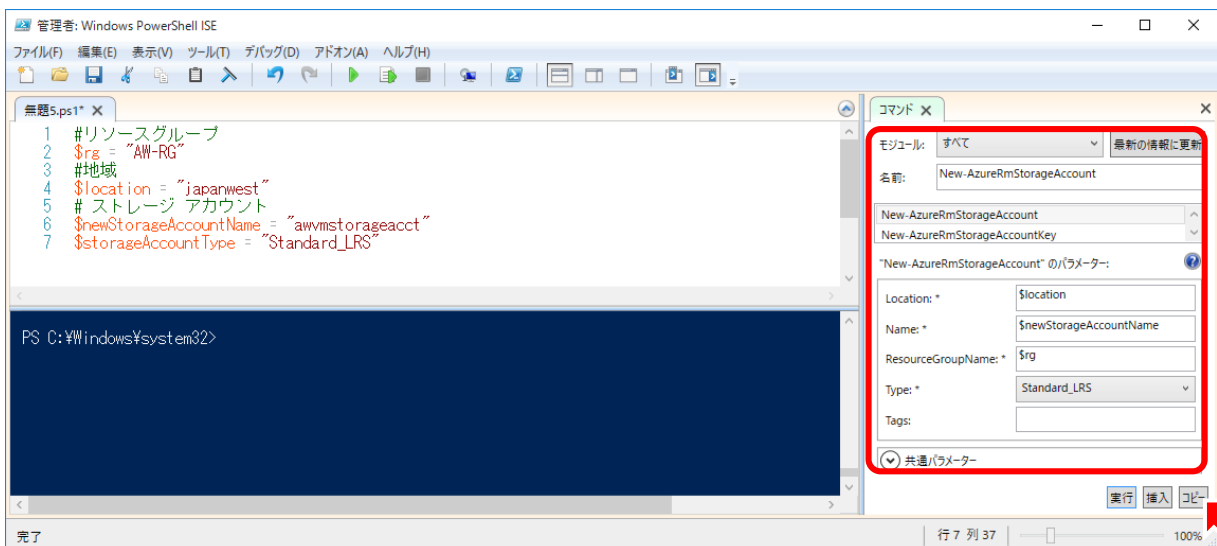
```
#リソースグループ
$rg = "AW-RG"
#地域
$location = "japanwest"
# ストレージ アカウント
$newStorageAccountName = "awvmstorageacct"
```

### ワンポイント

ストレージ アカウント名は Azure で一意な名前にする必要があります。既存のストレージ アカウント名と重複する場合、上記とは異なる名前に変更してください。

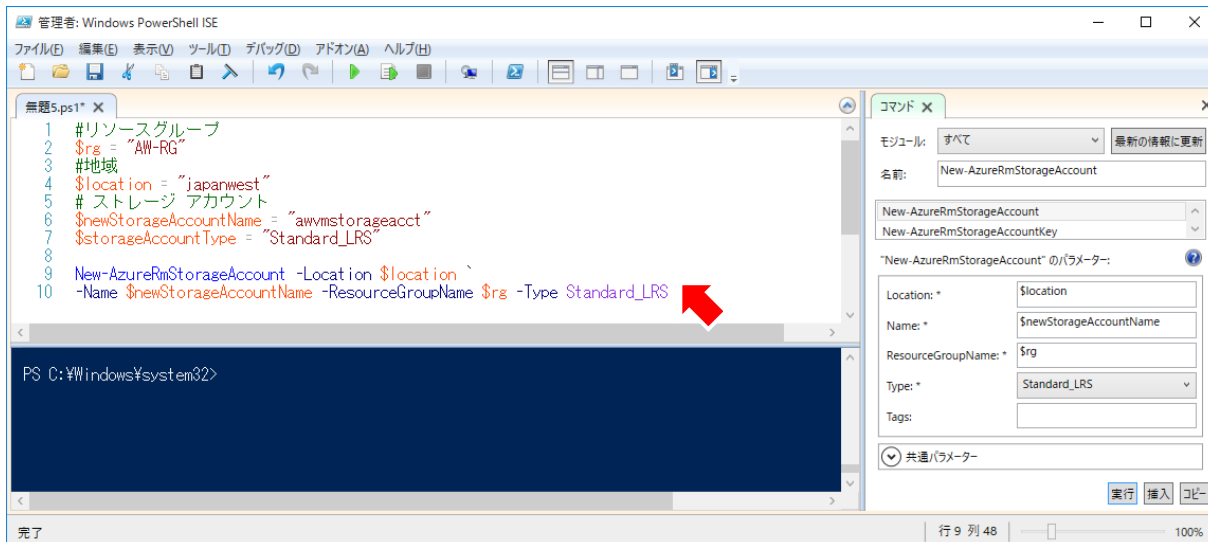
3. コマンド アドオン ウィンドウの [名前] に「**New-AzureRmStorageAccount**」と入力し、リストボックスに表示される「New-AzureRmStorageAccount」を選択します。[New-AzureRmStorageAccount のパラメーター] に下表の設定値を入力して [コピー] ボタンをクリックします。

パラメーター	設定値
Location	\$location
Name	\$newStorageAccountName
ResourceGroupName	\$rg
Type	Standard_LRS



4. スクリプト ウィンドウで、最終行に 2 回改行して、コピーしたコードを貼り付けます。この操作で、以下のコマンドが作成されたことを確認します。

```
New-AzureRmStorageAccount -Location $location `
-Name $newStorageAccountName -ResourceGroupName $rg -Type Standard_LRS
```



#### ワンポイント

8 行目と 9 行目は、1 行のコードとして生成されますが、読みやすくするため、8 行目の行末に「`」(行継続文字) を記述し、改行しています。

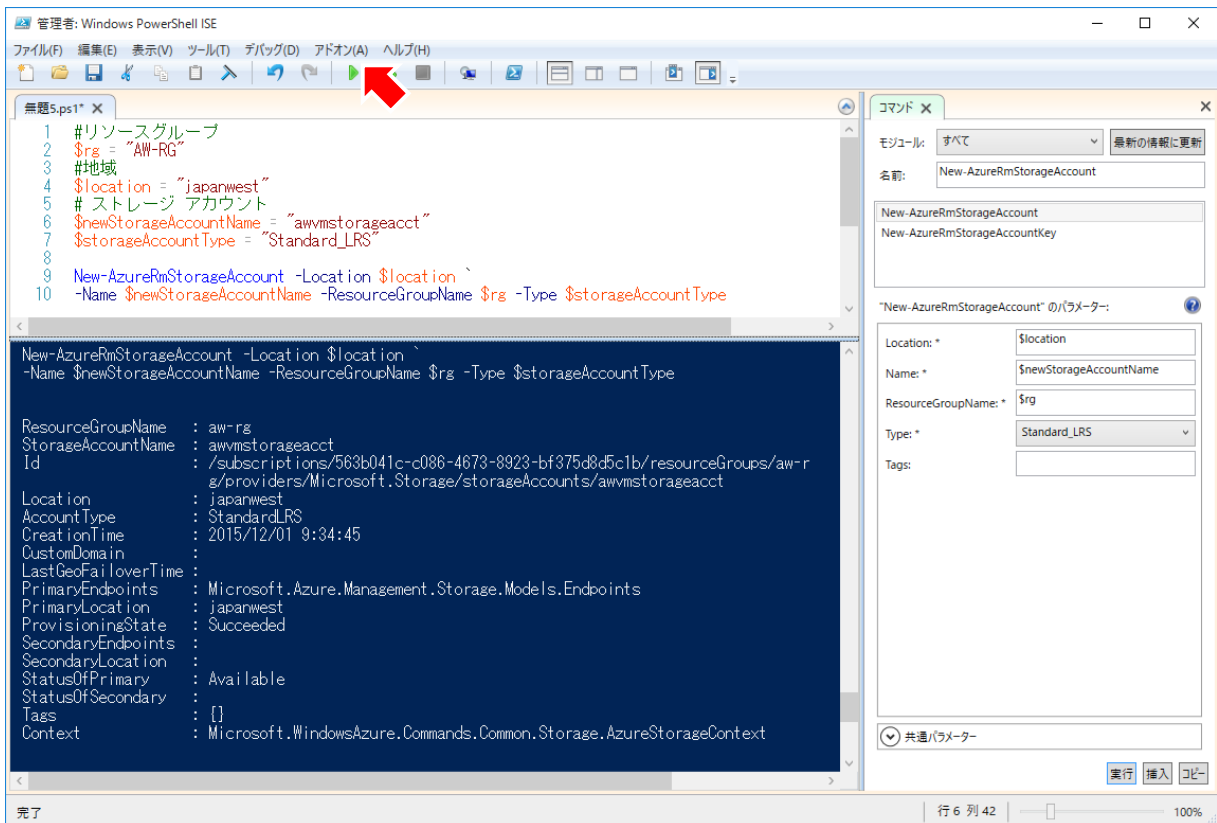
5. パラメーターのリストから渡された Type の値を変数化するため 変数宣言部の最終行に、改行を加えた後、以下の行を追加します。

```
$storageAccountType = "Standard_LRS"
```

6. New-AzureRmStorageAccount コマンドレットを以下に変更します。

```
New-AzureRmStorageAccount -Location $location `
-Name $newStorageAccountName -ResourceGroupName $rg -Type $storageAccountType
```

7. [スクリプトを実行] ボタンをクリックして実行し、ストレージ アカウントを作成します。



### ワンポイント

ストレージ アカウントの作成が完了するまで、数分かかります。このコマンドレットの実行により返される結果で、ProvisioningStatus が Succeeded になることを確認します。

8. Windows PowerShell ISE の [ファイル] メニューから [名前を付けて保存] を選択し、ここまで作成したスクリプトを作業用フォルダーに、「**Create-StorageAccount.ps1**」という名前で保存します。作成したスクリプトは、以下のコードになります。

```
#リソースグループ

$rg = "AW-RG"
#地域

$location = "japanwest"
# ストレージ アカウント

$newStorageAccountName = "awvmstorageacct"
$storageAccountType = "Standard_LRS"

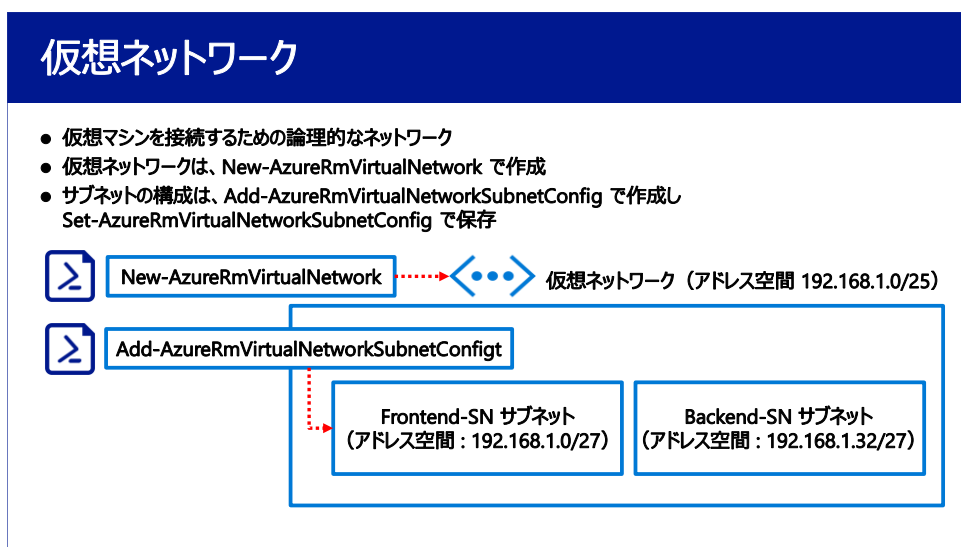
New-AzureRmStorageAccount -Location $location `
-Name $newStorageAccountName -ResourceGroupName $rg -Type $storageAccountType
```

9. [Create-StorageAccount.ps1] タブを閉じます。

### 3.3 仮想ネットワークとサブネットの作成

Azure 仮想ネットワークは、仮想マシンを接続するための論理的なネットワークです。1 つの仮想ネットワークに接続された仮想マシンは、お互い自由に通信することができます。仮想ネットワークのアドレス空間は、クラス A (10.0.0.0/8)、クラス B (172.16.0.0/12)、クラス C (192.168.0.0/16) のプライベート IP アドレスを設定することができます。また、仮想ネットワークには、複数のサブネットを作成することができます。

仮想ネットワークは、`New-AzureRmVirtualNetwork` で作成します。また、サブネット構成は、`Add-AzureRmVirtualNetworkSubnetConfig` で作成し、`Set-AzureRmVirtualNetworkSubnetConfig` で保存します。



ここでは前の手順で作成したリソースグループ「AW-RG」に仮想ネットワークを作成します。なお、この手順を行う前に、使用するサブスクリプションに作成されている仮想ネットワークを確認してください。既存の仮想ネットワークは、`Get-AzureRmVirtualNetwork` コマンドレットを実行することで確認できます。このコマンドの結果として、何も返らなければ仮想ネットワークは、存在していないことになります。下表のアドレス空間を持つ仮想ネットワークを作成します。

仮想ネットワーク名	アドレス空間
AW-VNET	192.168.1.0/25

仮想ネットワークには下表の 2 つのサブネットを定義します。

配置するサブネット名	アドレス空間
Frontend-SN	192.168.1.0/27
Backend-SN	192.168.1.32/27

既存の仮想ネットワークのアドレス空間と重複している場合、作成するアドレス空間を変更して実習を行ってください。

仮想ネットワークは、New-AzureRmVirtualNetwork で作成します。このコマンドレットの主要なパラメーターは、下表のとおりです。

パラメーター	説明	必須
<b>Location</b>	仮想マシンを作成する地域を入力します。	<input type="radio"/>
<b>Name</b>	仮想ネットワーク名を入力します。	<input type="radio"/>
<b>ResourceGroupName</b>	仮想ネットワーク名を作成するリソースグループ名を入力します。	<input type="radio"/>
<b>DnsServer</b>	DNS サーバーを指定します。	
<b>AddressPrefix</b>	仮想ネットワークに定義するアドレス空間を入力します。	
<b>Subnet</b>	仮想ネットワークに作成するサブネットを指定します。	
<b>Tag</b>	仮想ネットワークに設定するタグを指定します。	
<b>Force</b>	ユーザーの確認を必要としない場合、指定します。	

サブネットの構成が、あらかじめ作成されている場合、New-AzureRmVirtualNetwork を実行するときに、使用するサブネットを渡すことができますが、サブネット構成が存在しない場合は、先に仮想ネットワークを作成しておいて、後からサブネット構成を追加することもできます。サブネットの構成は、Add-AzureRmVirtualNetworkSubnetConfig で作成し、Set-AzureRmVirtualNetworkSubnetConfig で保存します。この 2 つのコマンドレットのパラメーターは、共通で下表のパラメータを指定します。

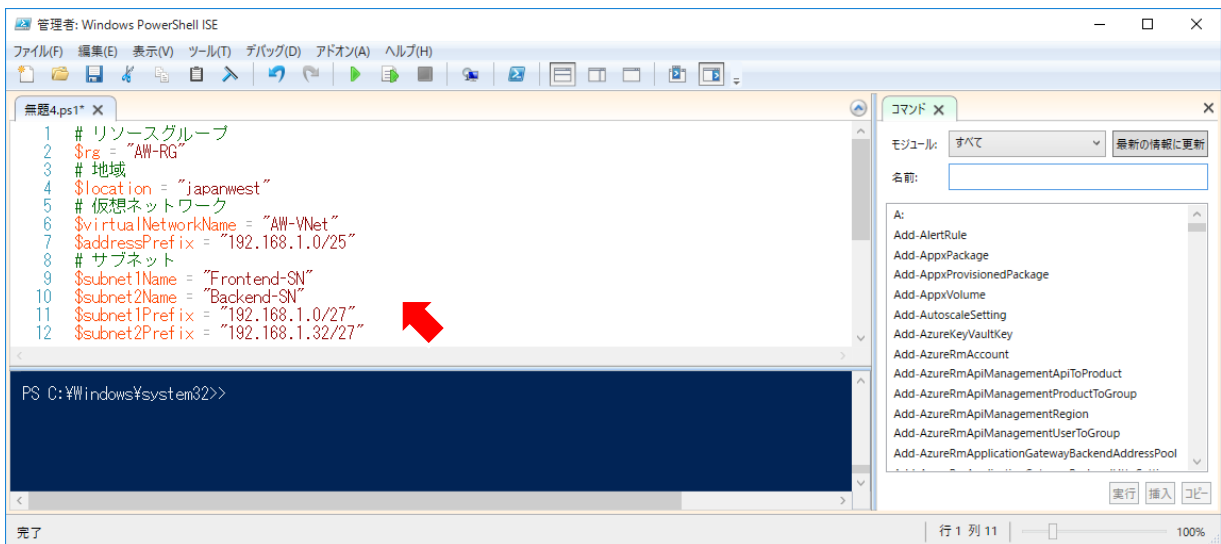
パラメーター	説明	必須
<b>AddressPrefix</b>	サブネットに定義するアドレス空間を入力します。	<input type="radio"/>
<b>Name</b>	サブネットの名前を入力します。	<input type="radio"/>
<b>VirtualNetwork</b>	サブネットを作成する仮想ネットワーク名を入力します。	<input type="radio"/>
<b>NetworkSecurityGroup</b>	ネットワーク セキュリティ グループを指定します。	

なお、サブネット構成の変更を仮想ネットワークに反映するには、最後に Set-AzureRmVirtualNetwork を実行する必要があります。このコマンドレットのパラメーターは、VirtualNetwork のみです。

パラメーター	説明	必須
<b>VirtualNetwork</b>	変更する仮想ネットワーク名を入力します。	<input type="radio"/>

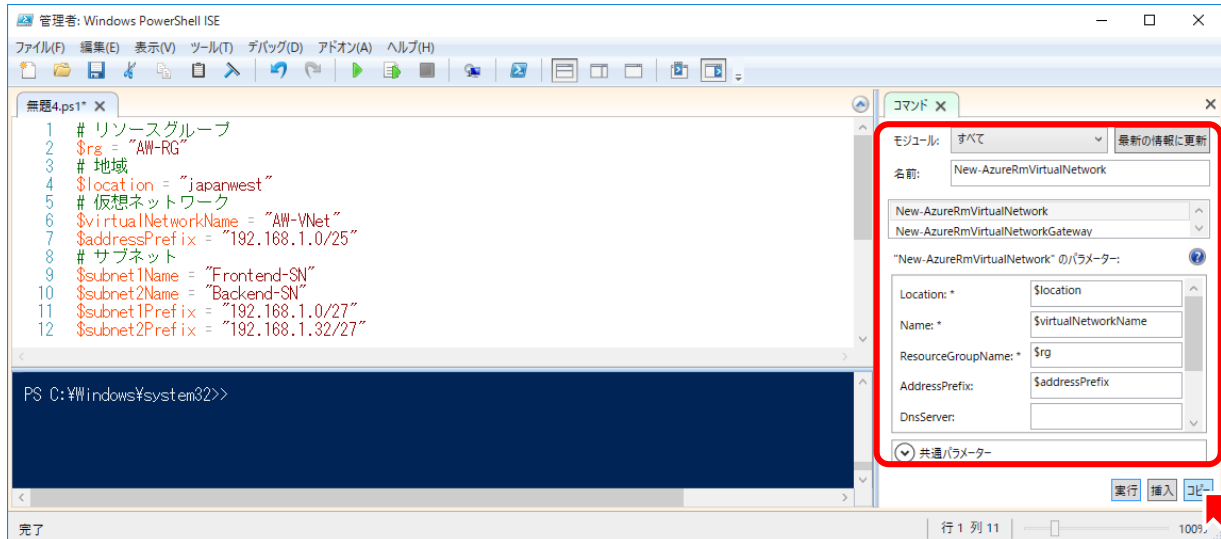
1. Windows PowerShell ISE の [表示] メニューの [スクリプト ウィンドウの表示] を選択して、新しいスクリプト ウィンドウを開きます。最初に以下の文字列変数宣言を記述しておきます。

```
# リソースグループ
$rg = "AW-RG"
# 地域
$location = "japanwest"
# 仮想ネットワーク
$virtualNetworkName = "AW-VNet"
$addressPrefix = "192.168.1.0/25"
# サブネット
$subnet1Name = "Frontend-SN"
$subnet2Name = "Backend-SN"
$subnet1Prefix = "192.168.1.0/27"
$subnet2Prefix = "192.168.1.32/27"
```



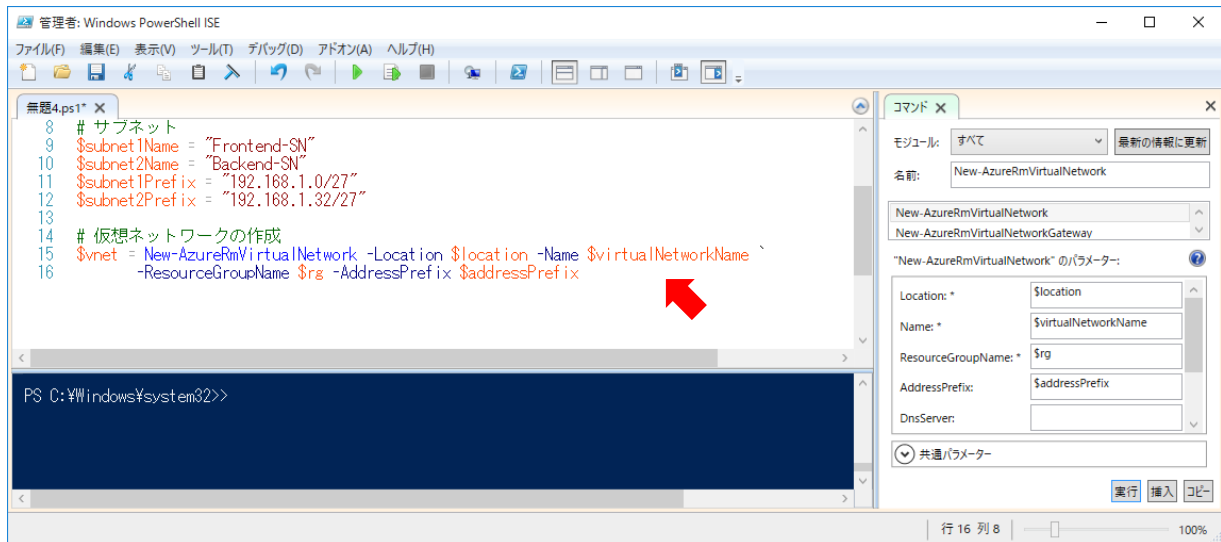
2. コマンド アドオン ウィンドウの [名前] に「**New-AzureRmVirtualNetwork**」と入力し、リストボックスに表示される「New-AzureRmVirtualNetwork」を選択します。[New-AzureRmVirtualNetwork のパラメーター] に下表の設定値を入力して [コピー] ボタンをクリックします。

パラメーター	設定値
<b>Location</b>	\$location
<b>Name</b>	\$virtualNetworkName
<b>ResourceGroupName</b>	\$rg
<b>AddressPrefix</b>	\$addressPrefix



3. スクリプト ウィンドウの最終行で、改行を 2 回入力したあと、コメント行として「# 仮想ネットワークの作成」と入力して改行し、「\$vnet = 」と入力した後、右クリックして、[貼り付け] を選択します。この操作で、以下のコマンドが作成されたことを確認します。

```
# 仮想ネットワークの作成
$vnet = New-AzureRmVirtualNetwork -Location $location -Name $virtualNetworkName `
-ResourceGroupName $rg -AddressPrefix $addressPrefix
```

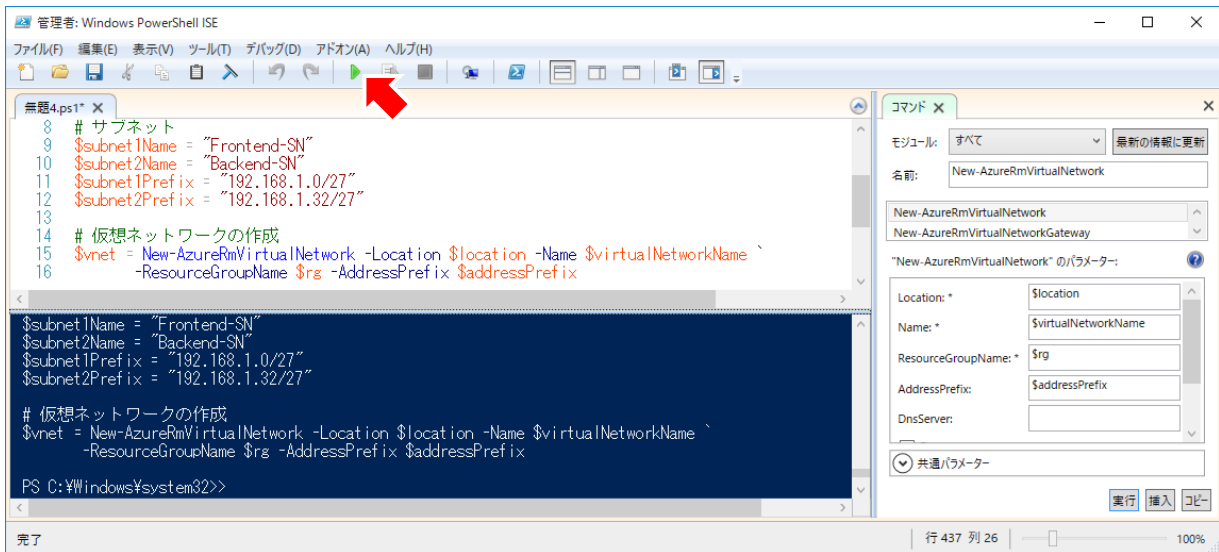


**ワンポイント**

読みやすくするため、\$virtualNetworkName の後に行継続文字「`」を入力して改行しています。  
PowerShell では、変数の宣言をしなくても変数を使用することができます。また、どんな型の値でも代入することができます。  
\$vnet には、作成された Azure 仮想ネットワーク リソースの情報がオブジェクトとして格納されます。

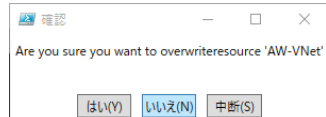


4. ここで、[スクリプトを実行] ボタンをクリックして実行し、仮想ネットワークを作成します。



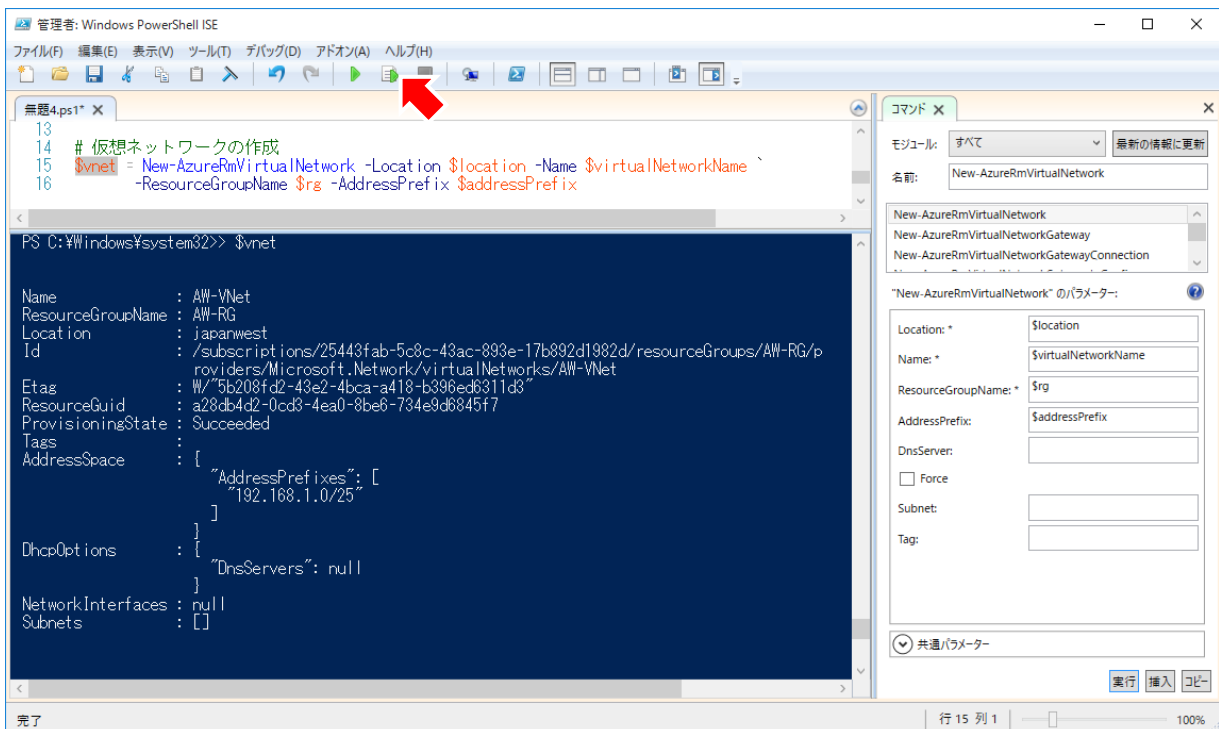
ワンポイント

既に、同じ名前の仮想ネットワークが存在する場合、以下のメッセージが表示されます。



既存の仮想ネットワークを変更してもよい場合は、[はい] をクリックします。既存の仮想ネットワークを残される場合は、\$virtualNetworkName 変数の値を「AW-VNet」ではない値に変更して実行してください。

5. New-AzureRmVirtualNetwork コマンドレットの先頭に書いた変数「\$vnet」を 選択状態にして、[選択項目を実行] ボタンで実行し、作成した仮想ネットワークの情報を確認します。

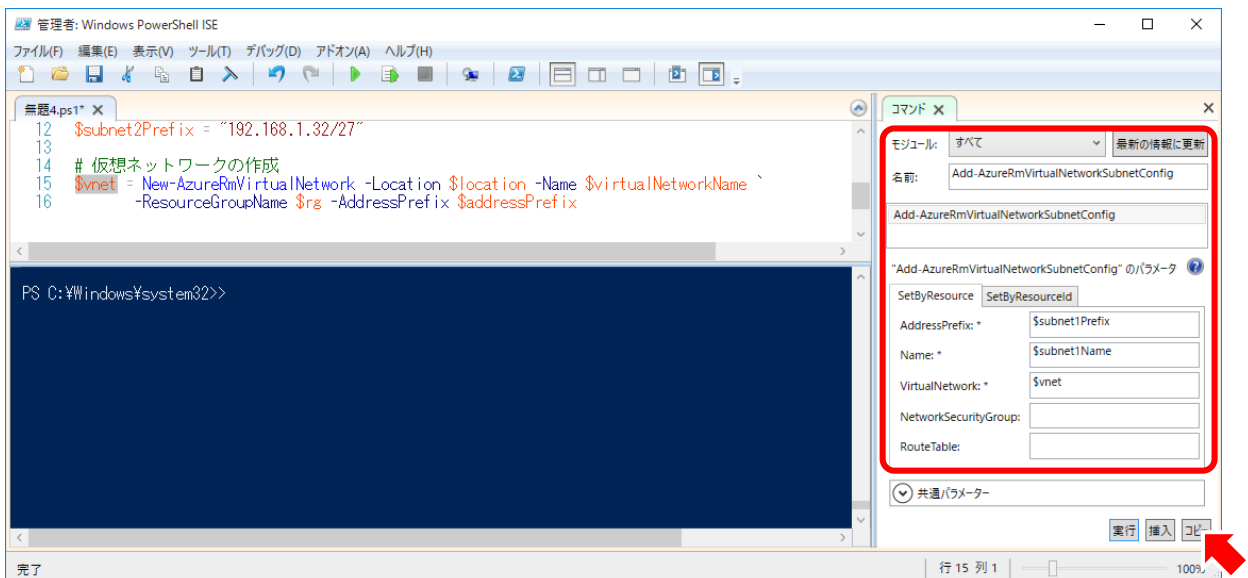


**ワンポイント**

このコマンドレットの実行により返される結果で、ProvisioningStatus が Succeeded になることを確認します。また、出力された情報の「Subnets」に着目します。「[]」と書かれていますが、角括弧内は配列を意味しています。配列の中身がないため、サブネットは、定義されていないことを意味しています。仮想ネットワークにサブネットを追加するには、Add-AzureRmVirtualNetworkSubnetConfig コマンドレットを使用します。

6. コマンド アドオン ウィンドウの [名前] に「**Add-AzureRmVirtualNetworkSubnetConfig**」と入力し、リストボックスに表示される「Add-AzureRmVirtualNetworkSubnetConfig」を選択します。[Add-AzureRmVirtualNetworkSubnetConfig のパラメーター] に下表の設定値を入力して [コピー] ボタンをクリックします。

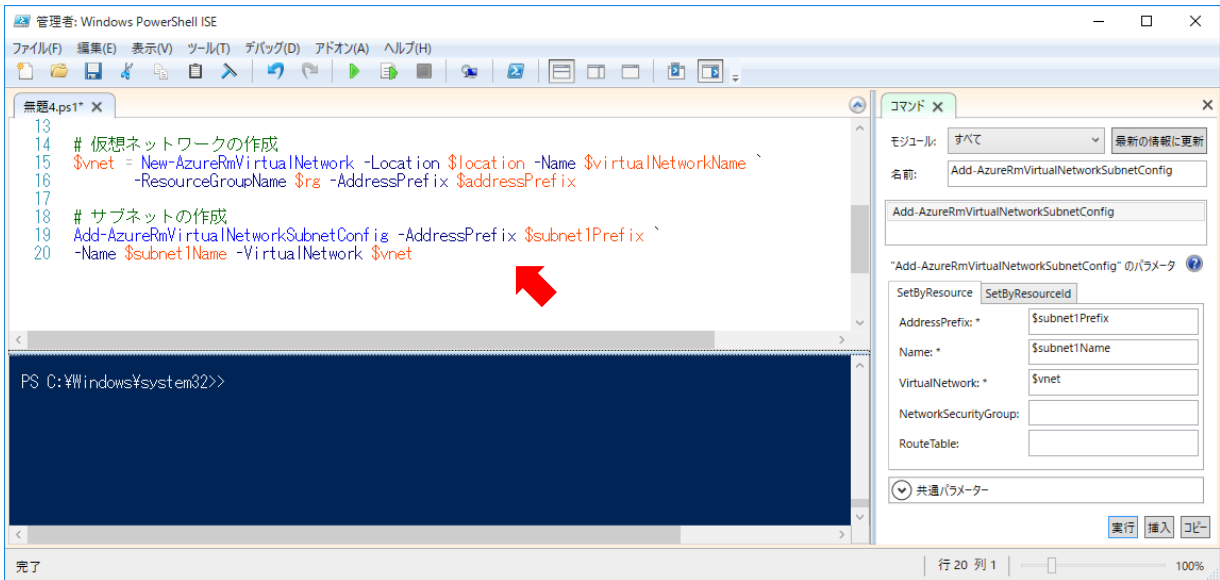
パラメーター	設定値
<b>AddressPrefix</b>	\$subnet1Prefix
<b>Name</b>	\$subnet1Name
<b>virtualNetwork</b>	\$vnet



7. スクリプト ウィンドウの最終行 2 回改行した後、コメント行として「# サブネットの作成」と入力して改行し、コピーしたコードを貼り付けます。この操作で、以下のコマンドが作成されたことを確認します。

```
# サブネットの作成

Add-AzureRmVirtualNetworkSubnetConfig -AddressPrefix $subnet1Prefix `
-Name $subnet1Name -VirtualNetwork $vnet
```

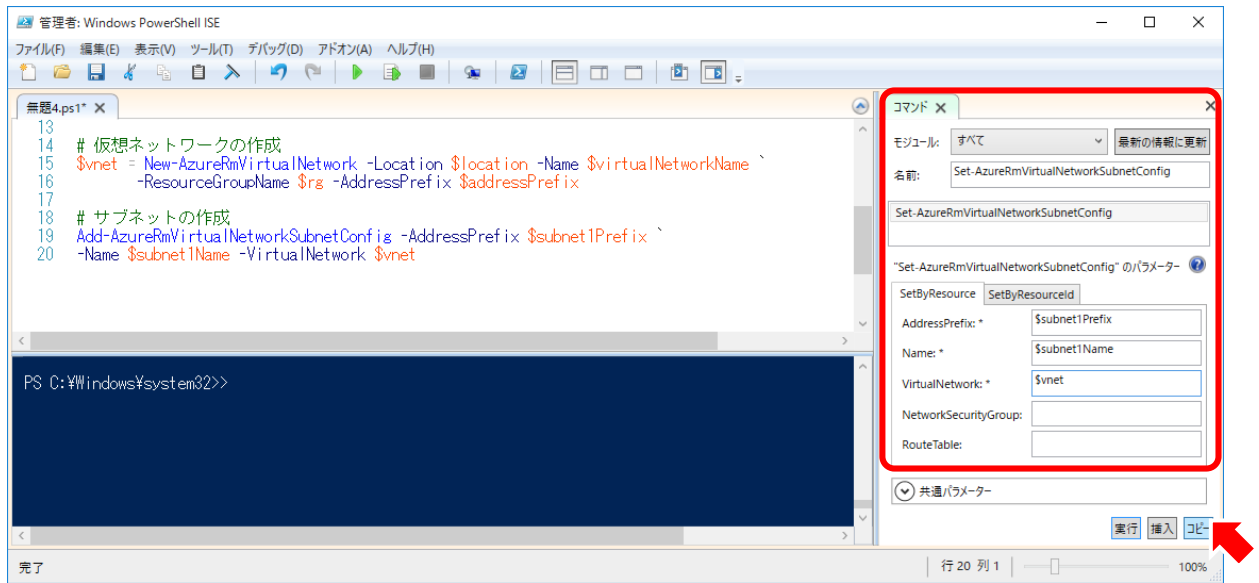


**ワンポイント**

Add-AzureRmVirtualNetworkSubnetConfig コマンドレットで追加されたサブネット定義を Azure に保存するには、Set-AzureRmVirtualNetworkSubnetConfig コマンドレットを実行します。

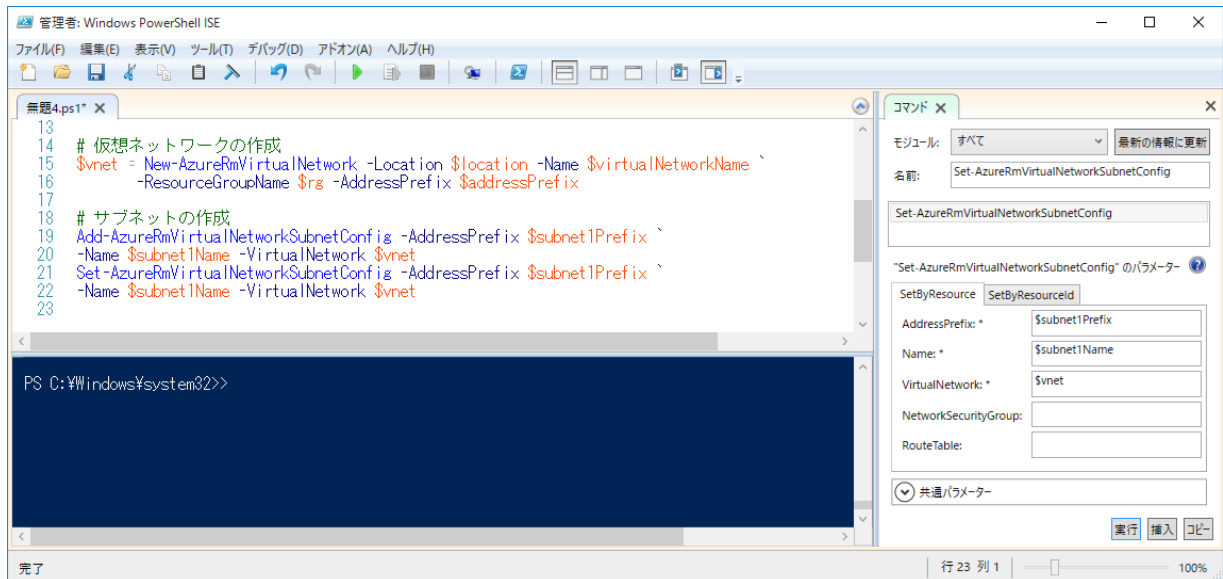
8. コマンド アドオン ウィンドウの [名前] に「**Set-AzureRmVirtualNetworkSubnetConfig**」と入力し、リストボックスに表示される「Set-AzureRmVirtualNetworkSubnetConfig」を選択します。[Set-AzureRmVirtualNetworkSubnetConfig のパラメーター] に下表の設定値を入力して [コピー] ボタンをクリックします。

パラメーター	設定値
<b>AddressPrefix</b>	\$subnet1Prefix
<b>Name</b>	\$subnet1Name
<b>virtualNetwork</b>	\$vnet



9. コピーしたコードを最終行に貼り付けます。この操作で、以下のコマンドが作成されたことを確認します。

```
Set-AzureRmVirtualNetworkSubnetConfig -AddressPrefix $subnet1Prefix `
-Name $subnet1Name -VirtualNetwork $vnet
```

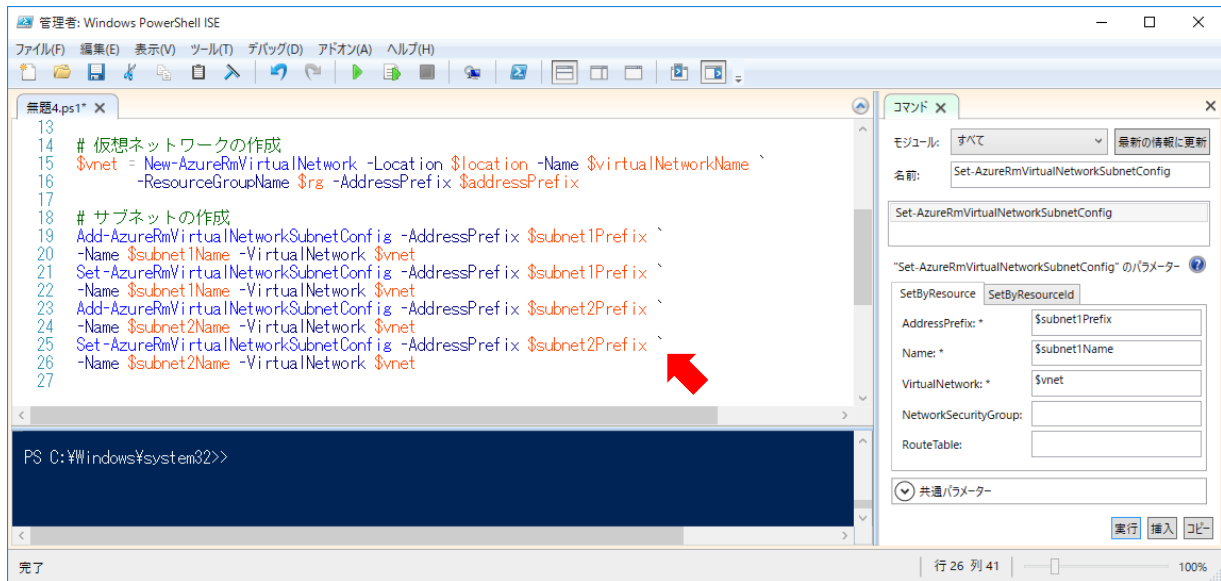


**ワンポイント**

仮想ネットワークには、2 種類のサブネットを定義する必要があります。ここまでの手順で作成した Add-AzureRmVirtualNetworkSubnetConfig と Set-AzureRmVirtualNetworkSubnetConfig コマンドレットのコードを再利用して、2 つ目のサブネットを定義します。

10. 前の手順で作成した Add-AzureRmVirtualNetworkSubnetConfig と Set-AzureRmVirtualNetworkSubnetConfig のコードをコピー アンド ペーストした後、下表を参照して、2 つ目のサブネットの定義としてしようできるように、変数を変更します。

最初のサブネットの変数名	2 つ目のサブネットの変数名
\$subnet1Prefix	\$subnet2Prefix
\$subnet1Name	\$subnet2Name

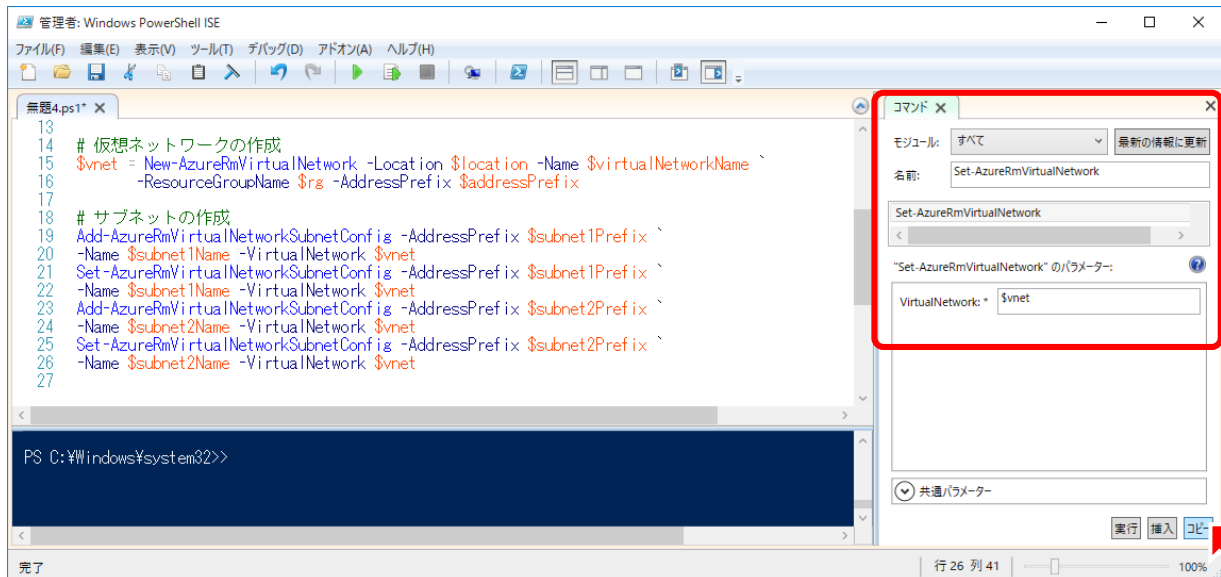


**ワンポイント**

ここまでのサブネット構成を仮想ネットワークに反映するため、最後に `Set-AzureRmVirtualNetwork` を実行します。

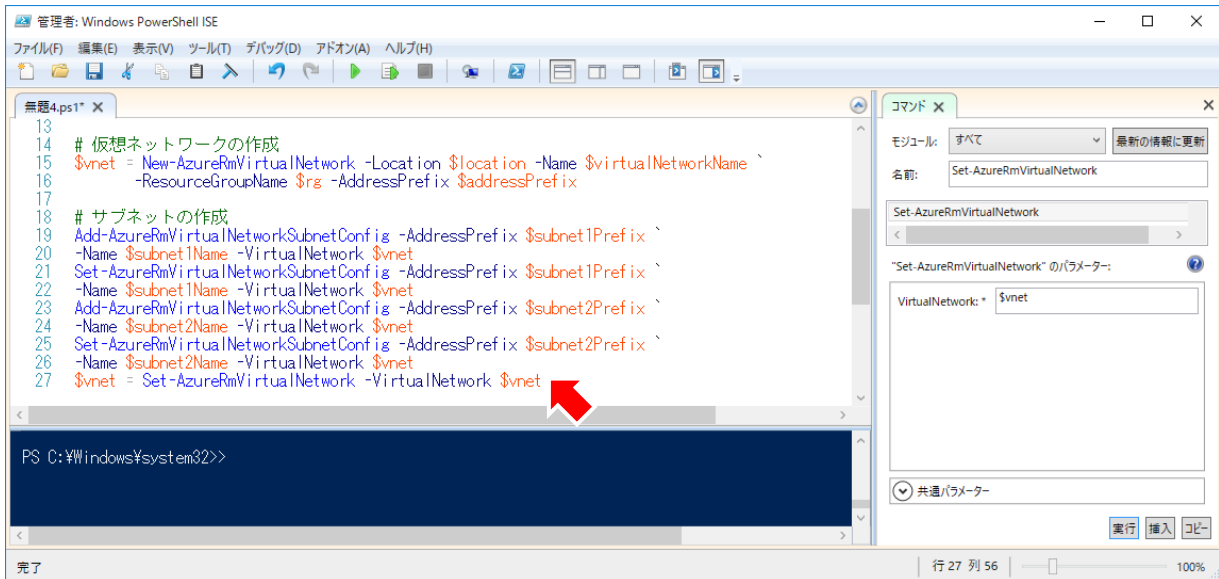
11. コマンド アドオン ウィンドウの [名前] に「**Set-AzureRmVirtualNetwork**」と入力し、リストボックスに表示される「Set-AzureRmVirtualNetwork」を選択します。[Set-AzureRmVirtualNetwork のパラメーター] に下表の設定値を入力して [コピー] ボタンをクリックします。

パラメーター	説明	設定値
<b>virtualNetwork</b>	変更を保存する仮想ネットワーク名を入力します。	\$vnet



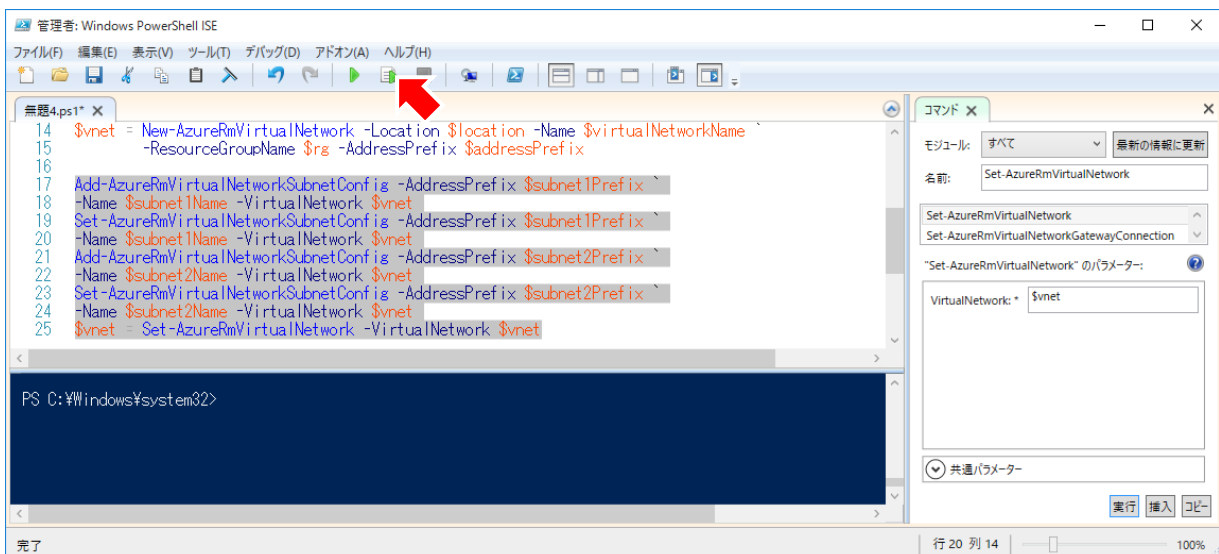
12. スクリプト ウィンドウで最終行に「**\$vnet =**」と入力した後、右クリックして、[貼り付け] を選択します。この操作で、以下のコマンドが作成されたことを確認します。

```
$vnet = Set-AzureRmVirtualNetwork -VirtualNetwork $vnet
```

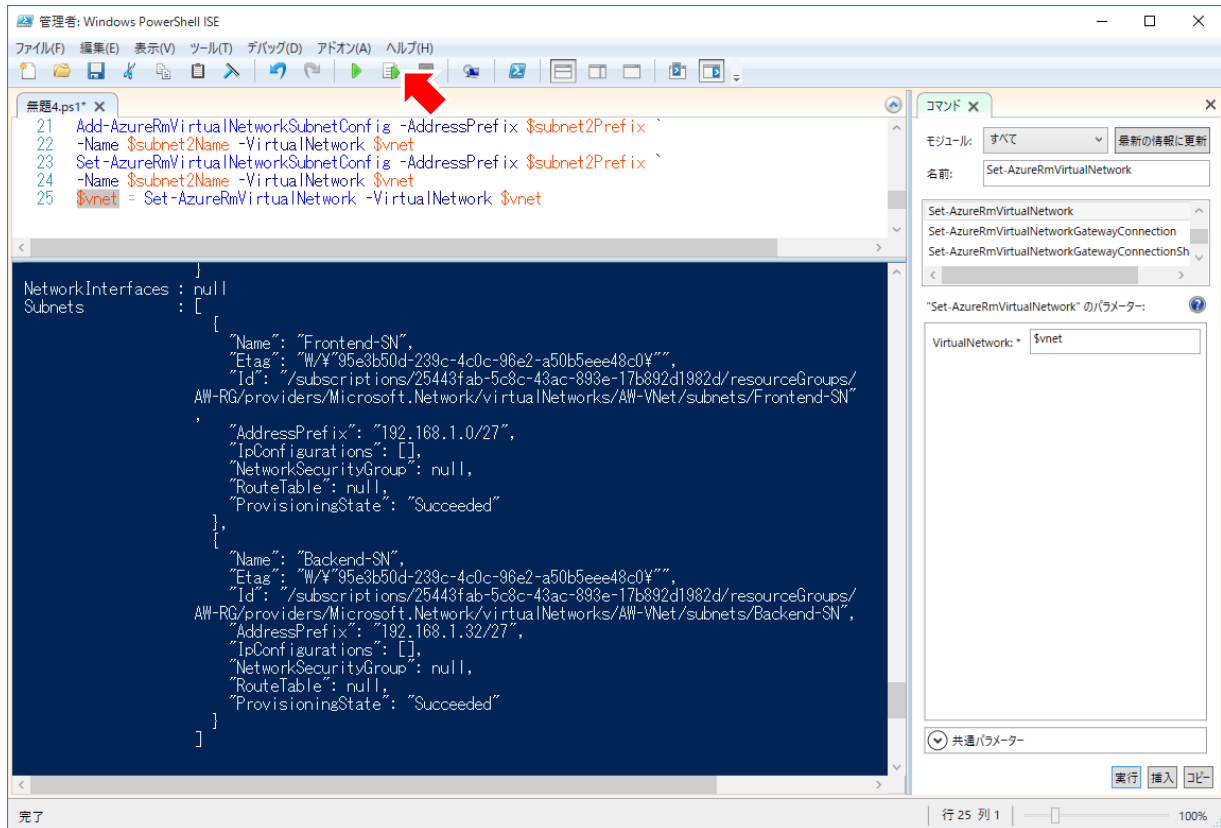


13. 仮想ネットワークに、2 つのサブネット構成を追加するため、以下のコードを選択状態にして、[選択項目を実行] ボタンをクリックします。

```
# サブネットの作成
Add-AzureRmVirtualNetworkSubnetConfig -AddressPrefix $subnet1Prefix `
-Name $subnet1Name -VirtualNetwork $vnet
Set-AzureRmVirtualNetworkSubnetConfig -AddressPrefix $subnet1Prefix `
-Name $subnet1Name -VirtualNetwork $vnet
Add-AzureRmVirtualNetworkSubnetConfig -AddressPrefix $subnet2Prefix `
-Name $subnet2Name -VirtualNetwork $vnet
Set-AzureRmVirtualNetworkSubnetConfig -AddressPrefix $subnet2Prefix `
-Name $subnet2Name -VirtualNetwork $vnet
$vnet = Set-AzureRmVirtualNetwork -VirtualNetwork $vnet
```



14. 作成した仮想ネットワークの情報を確認するため、Set-AzureRmVirtualNetwork コマンドレットの先頭に書いた「\$vnet」を選択状態にして、[選択項目を実行] ボタンをクリックします。



#### ワンポイント

仮想ネットワークの Subnets[] に 2 種類のサブネットが追加され、各サブネットの ProvisioningState が Succeeded になっていることを確認します。

15. Windows PowerShell ISE の [ファイル] メニューから [名前を付けて保存] を選択し、ここまで作成したスクリプトを作業用フォルダーに、「**Create-VNet.ps1**」という名前で保存します。作成したスクリプトは、以下のコードになります。

```

# リソースグループ
$rg = "AW-RG"
# 地域
$location = "japanwest"
# 仮想ネットワーク
$virtualNetworkName = "AW-VNet"
$addressPrefix = "192.168.1.0/25"
# サブネット
$subnet1Name = "Frontend-SN"
$subnet2Name = "Backend-SN"
$subnet1Prefix = "192.168.1.0/27"
$subnet2Prefix = "192.168.1.32/27"

# 仮想ネットワークの作成
    
```

```
$vnet = New-AzureRmVirtualNetwork -Location $location -Name $virtualNetworkName `
    -ResourceGroupName $rg -AddressPrefix $addressPrefix
```

# サブネットの作成

```
Add-AzureRmVirtualNetworkSubnetConfig -AddressPrefix $subnet1Prefix `
-Name $subnet1Name -VirtualNetwork $vnet
Set-AzureRmVirtualNetworkSubnetConfig -AddressPrefix $subnet1Prefix `
-Name $subnet1Name -VirtualNetwork $vnet
Add-AzureRmVirtualNetworkSubnetConfig -AddressPrefix $subnet2Prefix `
-Name $subnet2Name -VirtualNetwork $vnet
Set-AzureRmVirtualNetworkSubnetConfig -AddressPrefix $subnet2Prefix `
-Name $subnet2Name -VirtualNetwork $vnet
$vnet = Set-AzureRmVirtualNetwork -VirtualNetwork $vnet
```

16. [Create-VNet.ps1] タブを閉じます。

#### ワンポイント

作成した仮想ネットワークの構成は、Get-AzureRmVirtualNetwork を実行することで、後からいつでも確認することができます。なお、仮想ネットワークから不要になったサブネットを削除する場合は、Remove-AzureRmVirtualNetworkSubnetConfig コマンドレットを使用します。

```
Remove-AzureRmVirtualNetworkSubnetConfig -Name $subnet1Name `
-VirtualNetwork $vnet
```

このコマンドを実行するには、削除するサブネットには、事前に仮想マシンが配置されていない状態しておく必要があります。また、仮想ネットワークの削除では、Remove-AzureRmVirtualNetwork コマンドレットを使用します。

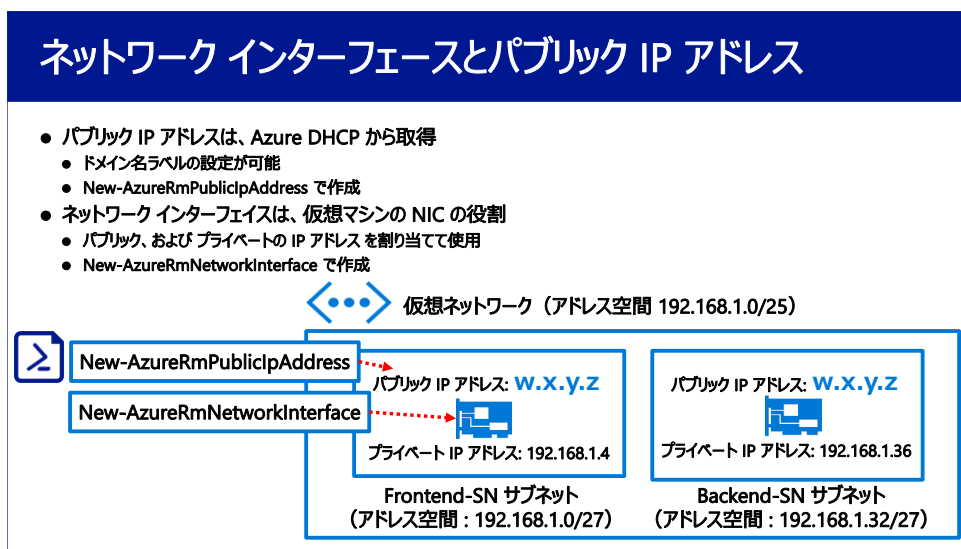
```
Remove-AzureRmVirtualNetwork -Name $vnet.Name -ResourceGroupName $rg
```

ここまでの手順では、Windows PowerShell ISE に慣れてもらうため、コマンド アドオン ウィンドウの機能を使用して、スクリプトを作成する手順を紹介してきましたが、以降では、作成するスクリプトを中心に解説していきます。



### 3.4 ネットワーク インターフェイスとパブリック IP アドレスの作成

ネットワーク インターフェイス リソースは、仮想マシンのネットワーク アダプタ (NIC) として機能するリソースです。IP アドレスやロードバランサーのアドレスプールの情報を割り当てることができます。自習書では、Web サーバーとデータベースサーバーの 2 台の仮想マシンを構成するため、ネットワーク インターフェイス リソースも、それぞれの仮想マシンで使用できるように 2 つ作成しておきます。



ネットワーク インターフェイス リソースには、仮想ネットワークでの通信に使用するプライベート IP アドレスと、グローバルインターネット環境からアクセスするためのパブリック IP アドレスを割り当てることができます。仮想ネットワークには、標準で Azure DHCP サーバーが用意されているため、サブネットに指定したアドレス空間の範囲内のプライベート IP アドレスと DNS サーバーのアドレスが動的に配布されます。実習では、プライベート IP アドレスを静的に割り当てます。なお、サブネットのアドレス空間の IP アドレスは、システムが使用する最初の 4 個 (0~3) と最後の 1 個 は使用できないことに注意してください。それ以外であれば、サブネットのアドレス空間内で未使用のアドレスを使用できますが、自習書では下表の内容で定義します。

仮想マシンの種類	仮想マシン名	配置するサブネット名	ネットワーク インターフェイス 名	プライベート IP アドレス
Web サーバー	Web01	Frontend-SN	Web01-NIC	192.168.1.4
データベース サーバー	SQL01	Backend-SN	SQL01-NIC	192.168.1.36

パブリック IP アドレス リソースは、New-AzureRmPublicIpAddress で作成します。このコマンドレットの主要なパラメーターは下表のとおりです。

パラメーター	説明	必須
<b>ResourceGroupName</b>	パブリック IP アドレスを作成するリソース グループ名を指定します。	○
<b>Location</b>	パブリック IP アドレスを作成する地域を指定します。	
<b>Name</b>	パブリック IP アドレスの名前を指定します。	
<b>DomainNameLabel</b>	パブリック IP アドレスのドメイン名ラベルを指定します。Azure で一意な名前です。	
<b>AllocationMethod</b>	パブリック IP アドレスの割り当て方法を Dynamic または Static から選択します。	○
<b>IdleTimeoutInMinutes</b>	アイドル タイムアウトの時間を分単位で指定します。	

<b>ReverseFqdn</b>	逆引きのための FQDN を指定します。	
<b>Tag</b>	パブリック IP アドレスに設定するタグを指定します。	
<b>Force</b>	ユーザーの確認を必要としない場合、指定します。	

自習書の手順では、パブリック IP アドレス リソースを作成する `New-AzureRmPublicIpAddress` で、下表のパラメーターと変数値を使用します。

パラメーター	変数名	値
<b>ResourceGroupName</b>	\$rg	AW-RG
<b>Location</b>	\$location	japanwest
<b>Name</b>	\$PublicIpAddressName	Web01-IP
<b>DomainNameLabel</b>	\$dnsNameforPublicIp	aw-we-ps
<b>AllocationMethod</b>	\$allocationMethod	Dynamic

1. [表示] メニューの [スクリプト ウィンドウの表示] を選択して、新しいスクリプト ウィンドウを開きます。
2. `New-AzureRmPublicIpAddress` のパラメーターに設定する変数と使用する値を記述します。

```
# リソースグループ
$rg = "AW-RG"
# 地域
$location = "japanwest"
# パブリックIPアドレス
$PublicIpAddressName = "Web01-IP"
$dnsNameforPublicIp = "aw-web-ps"
$allocationMethod = "Dynamic"
```

#### ワンポイント

本自習書では、Web サーバー用のパブリック IP アドレスに割り当てるドメイン名ラベルとして、「aw-web01-ps」を使用するように記載していますが、名前は、Azure で一意な名前を設定する必要があることに注意してください。実習を行う場合、「aw」の部分を別の文字列に置き換えるか任意の数値を付加するようにしてください。

3. 変数宣言の後、改行して以下のコードを記述します。

```
# パブリック IP アドレスの作成
$pip = New-AzureRmPublicIpAddress `
    -AllocationMethod $allocationMethod -ResourceGroupName $rg `
    -DomainNameLabel $dnsNameforPublicIP `
    -Location $location -Name $PublicIPAddressName
```

ネットワーク インターフェイスは、`New-AzureRmNetworkInterface` で作成します。このコマンドレットの主要なパラメーターは下表のとおりです。

パラメーター	説明	必須
<b>ResourceGroupName</b>	ネットワーク インターフェイスを作成するリソース グループ名を指定します。	○

<b>Location</b>	ネットワーク インターフェイスを作成する地域を指定します。	○
<b>Name</b>	ネットワーク インターフェイスの名前を指定します。	○
<b>Subnet</b>	ネットワーク インターフェイスを作成するサブネットを指定します。	○
<b>IpConfigurationName</b>	IP 構成の名前を指定します。	
<b>PrivateIpAddress</b>	ネットワーク インターフェイスに割り当てるプライベート IP アドレスを指定します。	
<b>PublicIpAddress</b>	ネットワーク インターフェイスに割り当てるパブリック IP アドレスを指定します。	
<b>NetworkSecurityGroup</b>	ネットワーク インターフェイスに割り当てるネットワーク セキュリティ グループを指定します。	
<b>LoadBalancerBackendAddressPool</b>	ロードバランサーを使用して負荷分散を行う場合、ロードバランサーのバックエンド アドレス プールを指定します。	
<b>LoadBalancerInboundNatRule</b>	ロードバランサーを使用して負荷分散を行う場合、ロードバランサーの NAT 受信規則を指定します。	
<b>Tag</b>	ネットワーク インターフェイスに設定するタグを指定します。	
<b>EnableIPForwarding</b>	IP ルーティングを有効化する場合、指定します。	
<b>Force</b>	ユーザーの確認を必要としない場合、指定します。	

#### ワンポイント

本自習書では、ロードバランサー、および ネットワーク セキュリティ グループの構成は含まれていません。ロードバランサー、および ネットワーク セキュリティ グループの構成は、「Microsoft Azure 自習書シリーズ Microsoft Azure を活用したスケール実現方法」を参照してください。

Subnet パラメーターに設定するサブネットは、ネットワーク インターフェイスの配置されるサブネットになりますが、この情報は、

Get-AzureRmVirtualNetworkSubnetConfig で設定します。このコマンドレットに使用するパラメーターは下表のとおりです。

パラメーター	説明	必須
<b>Name</b>	サブネットの名前	
<b>VirtualNetwork</b>	対象のサブネットがある仮想ネットワーク	○

また、Get-AzureRmVirtualNetworkSubnetConfig の VirtualNetwork パラメーターに設定する仮想ネットワークは、Get-AzureRmVirtualNetwork で取得します。このコマンドレットに使用するパラメーターは下表のとおりです。

パラメーター	説明	必須
<b>ResourceGroupName</b>	仮想ネットワークが作成されているリソース グループ名	
<b>Name</b>	仮想ネットワークの名前	

自習書の手順では、Web サーバー用のネットワーク インターフェイス リソースを作成する New-AzureRmNetworkInterface で、下表のパラメーターと変数値を使用します。

パラメーター	変数名	値
<b>ResourceGroupName</b>	\$rg	AW-RG

<b>Location</b>	\$location	Japanwest
<b>Name</b>	\$nicName	Web01-NIC
<b>Subnet</b>	\$subnet	Get-AzureRmVirtualNetworkSubnetConfig で取得
<b>PrivateIpAddress</b>	\$privateIpAddress	192.168.1.4
<b>PublicIpAddress</b>	\$pip	New-AzureRmPublicIpAddress で取得

サブネットの情報を取得するために Get-AzureRmVirtualNetworkSubnetConfig に下表のパラメーターと変数値を使用します。

パラメーター	変数名	値
<b>Name</b>	\$subnetName	Frontend-SN
<b>VirtualNetwork</b>	\$vnet	Get-AzureRmVirtualNetwork で取得

また、仮想ネットワークの情報を取得するために Get-AzureRmVirtualNetwork に下表のパラメーターと変数値を使用します。

パラメーター	変数名	値
<b>ResourceGroupName</b>	\$rg	AW-RG
<b>Name</b>	\$virtualNetworkName	AW-VNet

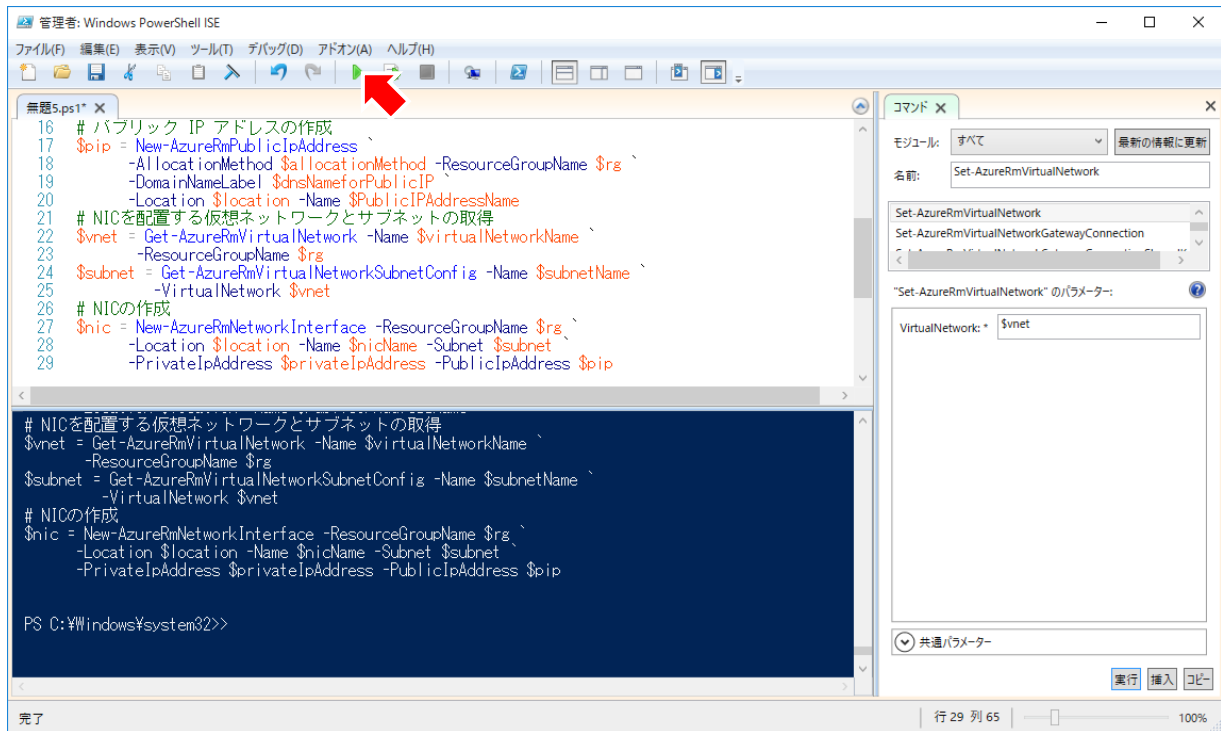
4. スクリプトの変数宣言部分に、ネットワーク インターフェイス リソースの作成で必要となる以下の変数宣言を追加します。

```
# 仮想ネットワーク
$virtualNetworkName = "AW-VNet"
$subnetName = "Frontend-SN"
# NIC
$nicName = "Web01-NIC"
$privateIpAddress = "192.168.1.4"
```

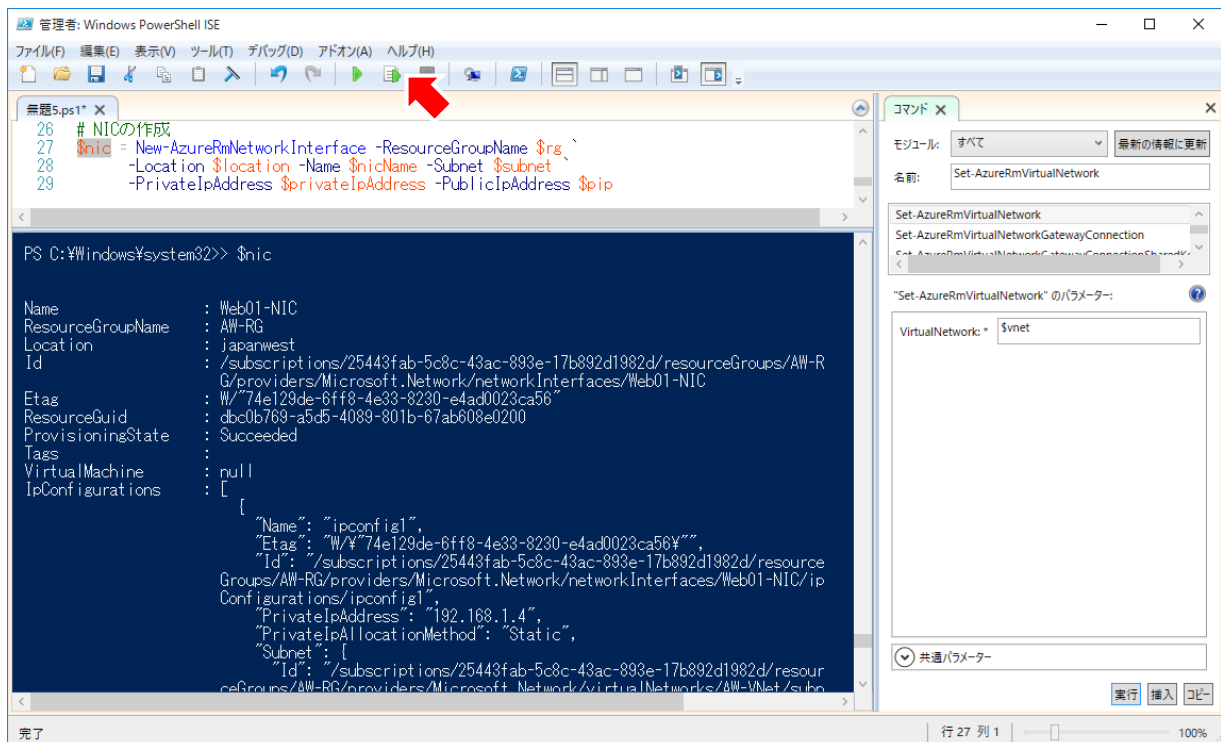
5. スクリプトの最終行に、以下のネットワーク インターフェイス リソース作成のためのコードを追加します。

```
# NIC を配置する仮想ネットワークとサブネットの取得
$vnet = Get-AzureRmVirtualNetwork -Name $virtualNetworkName `
    -ResourceGroupName $rg
$subnet = Get-AzureRmVirtualNetworkSubnetConfig -Name $subnetName `
    -VirtualNetwork $vnet
# NIC の作成
$nic = New-AzureRmNetworkInterface -ResourceGroupName $rg `
    -Location $location -Name $nicName -Subnet $subnet `
    -PrivateIpAddress $privateIpAddress -PublicIpAddress $pip
```

6. ここで、[スクリプトを実行] ボタンをクリックして実行し、Web サーバー用のネットワーク インターフェイス リソースを作成します。



7. New-AzureRmNetworkInterface コマンドレットの先頭に書いた「\$nic」を選択状態にして、[選択項目を実行] ボタンで実行し、作成したネットワーク インターフェイス リソースの情報を確認します。



#### ワンポイント

返される結果で、ProvisioningStatus が Succeeded になることを確認します。また、Ipconfigurations[] の配列で プラ

イベント IP アドレスとパブリック IP アドレスが設定されていることを確認します。

8. Windows PowerShell ISE の [ファイル] メニューから [名前を付けて保存] を選択し、ここまで作成したスクリプトを作業用フォルダーに、「**Create-NIC-for-Web01.ps1**」という名前で保存します。作成したスクリプトは、以下のコードになります。

```
#リソースグループ
$rg = "AW-RG"
#地域
$location = "japanwest"
# パブリックIPアドレス
$PublicIpAddressName = "Web01-IP"
$dnsNameforPublicIp = "aw-web-ps"
$allocationMethod = "Dynamic"
# 仮想ネットワーク
$virtualNetworkName = "AW-VNet"
$subnetName = "Frontend-SN"
# NIC
$nicName = "Web01-NIC"
$privateIpAddress = "192.168.1.4"

# パブリック IP アドレスの作成
$pip = New-AzureRmPublicIpAddress `
  -AllocationMethod $allocationMethod -ResourceGroupName $rg `
  -DomainNameLabel $dnsNameforPublicIP `
  -Location $location -Name $PublicIpAddressName
# NICを配置するサブネットの取得
$vnet = Get-AzureRmVirtualNetwork -Name $virtualNetworkName `
  -ResourceGroupName $rg
$subnet = Get-AzureRmVirtualNetworkSubnetConfig -Name $subnetName `
  -VirtualNetwork $vnet
# NICの作成
$nic = New-AzureRmNetworkInterface -ResourceGroupName $rg `
  -Location $location -Name $nicName -Subnet $subnet `
  -PrivateIpAddress $privateIpAddress -PublicIpAddress $pip
```

9. 次に、作成したスクリプトをデータベース サーバー用のネットワーク インターフェイス リソースの定義に変更するため、変数宣言部分を下表の値に修正します。

変数名	説明	Web サーバーのネットワーク インターフェイス	DB サーバーのネットワーク インターフェイス
\$PublicIpAddressName	パブリック IP アドレスの名前	Web01-IP	SQL01-IP
\$dnsNameforPublicIp	パブリック IP アドレスのドメイン名ラベル	aw-web-ps	aw-sql-ps
\$subnetName	サブネットの名前	Frontend-SN	Backend-SN
\$nicName	ネットワーク インターフェイスの名前	Web01-NIC	SQL01-NIC
\$privateIpAddress	ネットワーク インターフェイスに割り当てるプライベート IP アドレス	192.168.1.4	192.168.1.36

### ワンポイント

本自習書では、データベース サーバー用のパブリック IP アドレスに割り当てるドメイン名ラベルとして、「aw-sql01-ps」を使用するように記載していますが、名前は、Azure で一意な名前を設定する必要があることに注意してください。実習を行う場合、「aw」の部分の別の文字列に置き換えるか任意の数値を付加するようにしてください。

```
#リソースグループ
$rg = "AW-RG"
#地域
$location = "japanwest"
# パブリックIPアドレス
$PublicIpAddressName = "SQL01-IP"
$dnsNameforPublicIp = "aw-sql-ps"
$allocationMethod = "Dynamic"
# 仮想ネットワーク
$virtualNetworkName = "AW-VNet"
$subnetName = "Backend-SN"
# NIC
$nicName = "SQL01-NIC"
$privateIpAddress = "192.168.1.36"
```

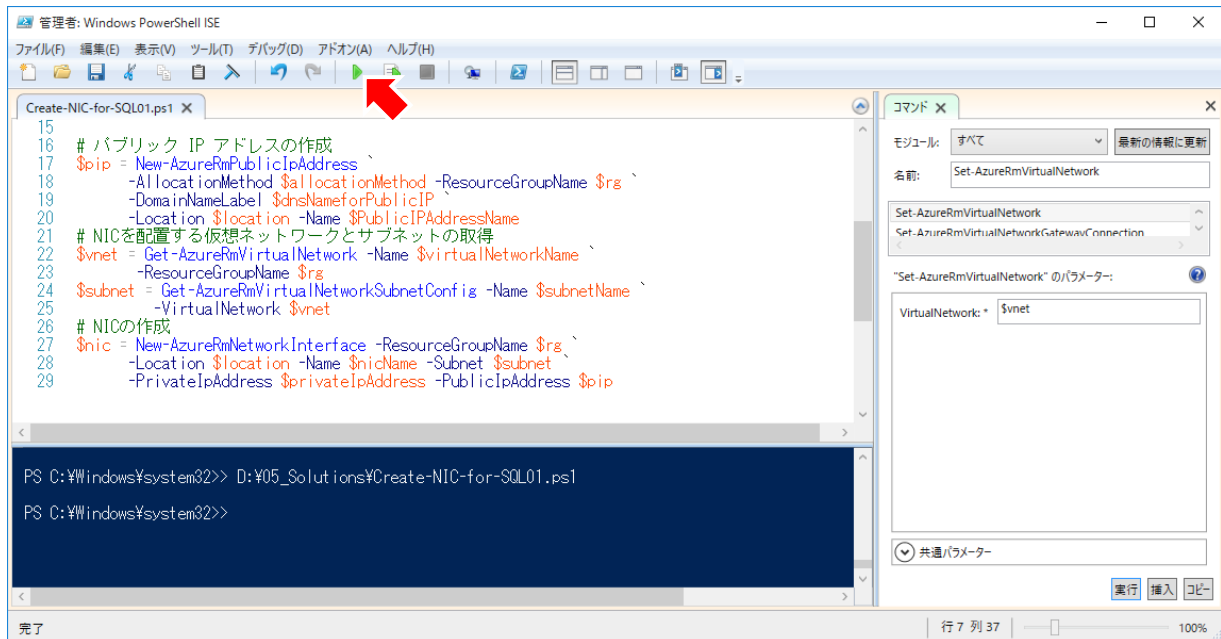
10. Windows PowerShell ISE の [ファイル] メニューから [名前を付けて保存] を選択し、Create-NIC-for-Web01.ps1 を修正したスクリプトを作業用フォルダーに、「**Create-NIC-for-SQL01.ps1**」という名前で保存します。作成したスクリプトは、以下のコードになります。

```
#リソースグループ
$rg = "AW-RG"
#地域
$location = "japanwest"
# パブリックIPアドレス
$PublicIpAddressName = "SQL01-IP"
$dnsNameforPublicIp = ""
$allocationMethod = "Dynamic"
# 仮想ネットワーク
$virtualNetworkName = "AW-VNet"
$subnetName = "Backend-SN"
# NIC
$nicName = "SQL01-NIC"
$privateIpAddress = "192.168.1.36"

# パブリック IP アドレスの作成
$pip = New-AzureRmPublicIpAddress `
    -AllocationMethod $allocationMethod -ResourceGroupName $rg `
    -DomainNameLabel $dnsNameforPublicIp `
    -Location $location -Name $PublicIpAddressName
# NICを配置するサブネットの取得
$vnet = Get-AzureRmVirtualNetwork -Name $virtualNetworkName `
    -ResourceGroupName $rg
$subnet = Get-AzureRmVirtualNetworkSubnetConfig -Name $subnetName `
    -VirtualNetwork $vnet
# NICの作成
$nic = New-AzureRmNetworkInterface -ResourceGroupName $rg `
    -Location $location -Name $nicName -Subnet $subnet `
    -PrivateIpAddress $privateIpAddress -PublicIpAddress $pip
```



11. ここで、[スクリプトを実行] ボタンをクリックして実行し、データベース サーバー用のネットワーク インターフェイス リソースを作成します。



12. [Create-NIC-for-SQL01.ps1] タブを閉じます。

### ワンポイント

ここまでの手順で作成したリソースを確認するには、Get-AzureRmResource コマンドレットを使用します。

```
Get-AzureRmResource | ? { $_.ResourceGroupName -eq $rg } | Select Name, ResourceType
```

```
PS C:\Windows\system32> Get-AzureRmResource | ? { $_.ResourceGroupName -eq $rg } | Select Name, ResourceType
Name                               ResourceType
-----                               -
SQL01-NIC                          Microsoft.Network/networkInterfaces
Web01-NIC                          Microsoft.Network/networkInterfaces
SQL01-IP                            Microsoft.Network/publicIPAddresses
Web01-IP                            Microsoft.Network/publicIPAddresses
AW-VNet                             Microsoft.Network/virtualNetworks
awvmstorageacct                    Microsoft.Storage/storageAccounts
```

なお、作成した ネットワーク インターフェイスを削除するには、Remove-AzureRmNetworkInterface コマンドレットを使用します。

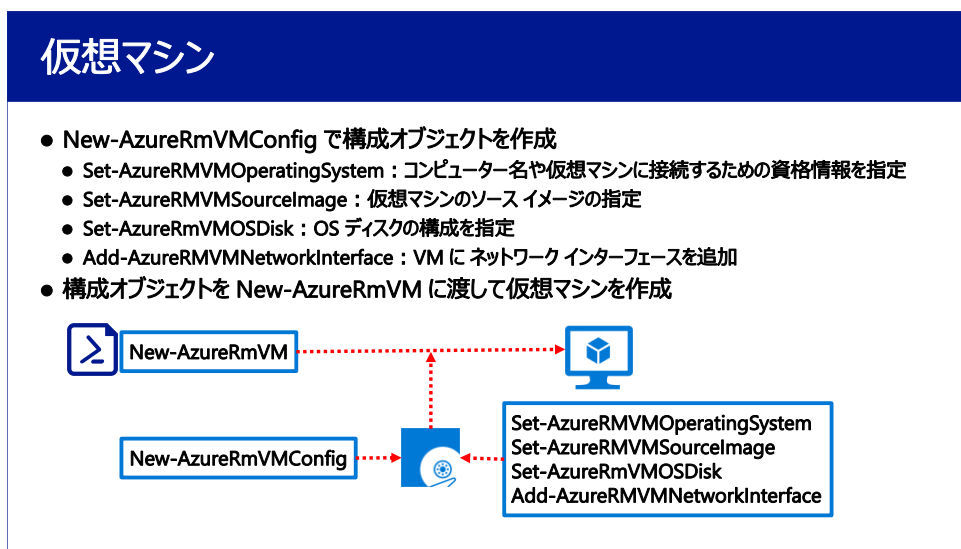
```
Remove-AzureRmNetworkInterface -Name $nicName -ResourceGroupName $rg
```

また、作成した パブリック IP を削除するには、Remove-AzureRmPublicIpAddress コマンドレットを使用します。

```
Remove-AzureRmPublicIpAddress -Name $PublicIPAddressName `
-ResourceGroupName $rg
```

### 3.5 仮想マシンの作成

仮想マシンの作成は、New-AzureRmVM コマンドレットを使用します。仮想マシン リソースには、管理者の資格情報、マシン サイズ、ディスク構成、ネットワーク インターフェイスなど様々なパラメーターを設定する必要があります。これらの構成情報は、New-AzureRmVM で必須の VM パラメーターに渡す仮想マシンの構成オブジェクトで設定します。仮想マシンの構成オブジェクトは、New-AzureRmVMConfig で作成します。



ここでは、Web サーバーとデータベース サーバーの 2 台の仮想マシンを作成します。New-AzureRmVM のパラメーターは下表のとおりです。

パラメーター	説明	必須
<b>ResourceGroupName</b>	仮想マシンを作成するリソース グループ名を指定します。	○
<b>Location</b>	仮想マシンを作成する地域を指定します。	○
<b>VM</b>	仮想マシンを構成するためのオブジェクトを指定します。	○
<b>Tags</b>	仮想マシンに設定するタグを指定します。	

仮想マシン構成オブジェクトを作成するための New-AzureRmVMConfig に指定するパラメーターは下表のとおりです。

パラメーター	説明	必須
<b>VMName</b>	仮想マシンの名前を指定します。	○
<b>VMSize</b>	仮想マシンのサイズを指定します。	○
<b>AvailabilitySetId</b>	仮想マシンの可用性セットの ID を指定します。	

### ワンポイント

VMSize パラメーターに設定できる値は、Get-AzureRmVMSize で取得することができます。

```
PS C:\Windows\system32> $location = "japanwest"
Get-AzureRmVMSize -Location $location |
select Name, MemoryInMB, NumberOfCores, ResourceDiskSizeInMB

Name           MemoryInMB  NumberOfCores  ResourceDiskSizeInMB
-----
Standard_A0    768         1              20480
Standard_A1    1792        1              71680
Standard_A2    3584        2              138240
Standard_A3    7168        4              291840
Standard_A5    14336       2              138240
Standard_A4    14336       8              619520
Standard_A6    28672       4              291840
Standard_A7    57344       8              619520
Basic_A0       768         1              20480
Basic_A1       1792        1              40960
Basic_A2       3584        2              61440
Basic_A3       7168        4              122880
Basic_A4       14336       8              245760
Standard_DS1   3584        1              7168
Standard_DS2   7168        2              28762
Standard_DS3   14336       4              28762
Standard_DS4   28672       8              57244
Standard_DS11  14336       2              28872
Standard_DS12  28672       4              57344
Standard_DS13  57344       8              114688
Standard_DS14  114688      16             229376
Standard_D1    3584        1              51200
Standard_D2    7168        2              102400
Standard_D3    14336       4              204800
Standard_D4    28672       8              409600
Standard_D11   14336       2              102400
Standard_D12   28672       4              204800
Standard_D13   57344       8              409600
Standard_D14   114688      16             819200
```

仮想マシンのサイズと価格については、「[Virtual Machines の価格](#)」を参照してください。

New-AzureRmVMConfig により、仮想マシン名とインスタンスのサイズが指定された仮想マシンの構成オブジェクトが作成されます。

プロビジョニングに必要なその他の構成情報は下表に掲載した複数のコマンドレットを使用して設定します。いずれのコマンドレットのパラメーターにも VM 構成オブジェクトは、必須で要求されます。

構成オブジェクトを設定するコマンドレット	説明
<b>Set-AzureRMVMOperatingSystem</b>	コンピューター名や仮想マシンに接続するための資格情報などの仮想マシンのプロパティを指定します。
<b>Set-AzureRMVMSourceImage</b>	仮想マシンを作成するためのソースとなるイメージを指定します。ソース イメージは、Offer、PublisherName、SKU、Version の 4 つの情報により、Azure のギャラリーから取得することができます。
<b>Set-AzureRmVMOSDisk</b>	OS ディスクの配置先として使用する Blob ストレージのパスなどを設定します。
<b>Add-AzureRMVMNetworkInterface</b>	仮想マシンに接続するネットワーク インターフェイスの ID を指定します。

Windows Server の仮想マシンを作成する場合、Set-AzureRMVMOperatingSystem に設定する主なパラメーターは下表のとおりです。

パラメーター	説明	必須
<b>ComputerName</b>	コンピューター名を指定します。	○
<b>Credential</b>	仮想マシンに作成される管理者アカウントのための資格情報を指定します。	○
<b>VM</b>	設定する仮想マシン構成オブジェクトを指定します。	○
<b>Windows</b>	Windows OS であることを指定する場合、記述します。Linux の場合は、Linux パラメーターが必須です。	○
<b>EnableAutoUpdate</b>	自動更新を有効化する場合、指定します。	
<b>ProvisionVMAgent</b>	仮想マシンに VM エージェントをインストールする場合、指定します。	
<b>CustomData</b>	カスタム データを記述する場合、指定します。	
<b>TimeZone</b>	仮想マシンのタイムゾーンを指定します。	

Azure 仮想マシン エージェント (VM エージェント) は、Azure 仮想マシン拡張機能のインストール、構成、管理、および実行に使用されます。VM 拡張機能により、Microsoft とその他のサード パーティが提供する動的機能を使用できるようになります。この自習書では、カスタム スクリプト拡張機能と DSC (Desired State Configuration) 拡張機能を使用した IIS のインストールを手順に含めています。

Set-AzureRMVMSourceImage のパラメーターは、すべて必須で下表のとおりです。

パラメーター	説明	必須
<b>PublisherName</b>	仮想マシン イメージの発行元の名前を指定します。	○
<b>Offer</b>	仮想マシン イメージのオファー名を指定します。	○
<b>Skus</b>	仮想マシン イメージの SKU を指定します。	○
<b>Version</b>	仮想マシン イメージのバージョンを指定します。	○
<b>VM</b>	設定する仮想マシン構成オブジェクトを指定します。	○

### ワンポイント

現在、Azure で提供される仮想マシン イメージは、様々な提供元にわたります。この情報は、Get-AzureRmVMImagePublisher で取得することができます。プロビジョニングする提供元の情報を Set-AzureRmVMSourceImage の PublisherName パラメーターに設定する必要があります。

```
PS C:\Windows\system32> $location = "japanwest"
Get-AzureRmVMImagePublisher -Location $location

-----
PublisherName                               Location  Id
-----
4psa                                         japanwest /Subscriptions/25443fab-5c...
4ward365                                     japanwest /Subscriptions/25443fab-5c...
7isolutions                                 japanwest /Subscriptions/25443fab-5c...
a10networks                                 japanwest /Subscriptions/25443fab-5c...
abiquo                                       japanwest /Subscriptions/25443fab-5c...
active-navigation                          japanwest /Subscriptions/25443fab-5c...
activeeon                                    japanwest /Subscriptions/25443fab-5c...
adam-software                               japanwest /Subscriptions/25443fab-5c...
adatao                                       japanwest /Subscriptions/25443fab-5c...
adobe                                        japanwest /Subscriptions/25443fab-5c...
adobe_test                                  japanwest /Subscriptions/25443fab-5c...
advantech                                   japanwest /Subscriptions/25443fab-5c...
advantech-webaccess                        japanwest /Subscriptions/25443fab-5c...
aerospike                                   japanwest /Subscriptions/25443fab-5c...
aimsinnovation                             japanwest /Subscriptions/25443fab-5c...
aiscaler-cache-control-ddos-and-url-rewriting
alchisoft                                   japanwest /Subscriptions/25443fab-5c...
alertlogic                                  japanwest /Subscriptions/25443fab-5c...
```

PublisherName に「Microsoft」が含まれるデータ行のみを取得するには、以下のコマンドを実行します。

```
Get-AzureRmVMImagePublisher -Location $location `
| ?{$_ .PublisherName -like "Microsoft*"}
```

```
PS C:\Windows\system32> $location = "japanwest"
Get-AzureRmVMImagePublisher -Location $location `
| ?{$_ .PublisherName -like "Microsoft*"}
```

PublisherName	Location	Id
Microsoft	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft-ads	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.Azure.Applications	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.Azure.Backup.Test	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.Azure.Diagnostics	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.Azure.Extensions	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.Azure.RecoveryServices	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.Azure.Security	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.Azure.Security.Internal	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.Azure.WindowsFabric.Test	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.AzureCAT.AzureEnhancedMonitoring	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.AzureCAT.AzureEnhancedMonitoringTest	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.Compute	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.EnterpriseCloud.Monitoring	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.EnterpriseCloud.Monitoring.Test	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.HpcCompute	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.HpcPack	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.OSTCExtensions	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.Powershell	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.Powershell.Install	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.Powershell.Internal	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.Powershell.Internal.Telemetry	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.Powershell.PaaS	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.Powershell.Test	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.Powershell.Test2	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.Powershell.Test3	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.Powershell.URCTest	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.Powershell.Wmf	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.Powershell.Wmf4Test	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.Powershell.Wmf5	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.SqlServer.Managability.IaaS.Test	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.SqlServer.Management	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.SystemCenter	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.VisualStudio	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.VisualStudio.Azure.ETWTraceListenerService	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.VisualStudio.Azure.RemoteDebug	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.VisualStudio.Azure.RemoteDebug.Json	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.VisualStudio.ServiceProfiler	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.VisualStudio.ServiceProfiler.Test	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.Windows.AzureRemoteApp.Test	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.Windows.RemoteDesktop	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.WindowsAzure.Compute	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.AzureSiteRecovery	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.BizTalkServer	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.DynamicsAX	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.DynamicsNAV	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.HybridCloudStorage	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.SharePoint	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.SQLServer	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.VisualStudio	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.WindowsServer	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.WindowsServerEssentials	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.WindowsServerHPCPack	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...
Microsoft.WindowsServerRemoteDesktop	japanwest	/Subscriptions/25443fab-5c8c-43ac-893e-...

IIS だけを使用するシンプルな Windows Server OS 仮想マシンを作成する場合、PublisherName が「MicrosoftWindowsServer」の仮想マシンイメージを選択します。SQL Server がインストールされる Windows Server OS 仮想マシンを作成する場合は、PublisherName が「MicrosoftSQLServer」の仮想マシンイメージを選択します。

Offer に設定できる値は、Get-AzureRmVMImageOffer で取得することができます。例えば PublisherName が「MicrosoftWindowsServer」の仮想マシンイメージで選択できるオファーを表示するには、以下のコマンドを実行します。

```
Get-AzureRmVMImageOffer -Location $location `
-PublisherName MicrosoftWindowsServer
```

```
Get-AzureRmVMImageOffer -Location $location `
-PublisherName MicrosoftWindowsServer

Offer          PublisherName      Location  Id
-----
WindowsServer MicrosoftWindowsServer japanwest /Subscriptions/25443fab-5c8c-43ac-893e-17b892d1982d/Pr...
```

Skus に設定できる値は、Get-AzureRmVMImageSkus で取得することができます。

例えば PublisherName が「MicrosoftWindowsServer」で Offer が「WindowsServer」の仮想マシンイメージで選択できるすべての SKU を表示するには、以下のコマンドを実行します。

```
Get-AzureRmVMImageSkus -Location $location -Offer WindowsServer `
-PublisherName MicrosoftWindowsServer
```

```
Get-AzureRmVMImageSkus -Location $location -Offer WindowsServer `
-PublisherName MicrosoftWindowsServer

Skus          Offer          PublisherName      Location  Id
-----
2008-R2-SP1   WindowsServer MicrosoftWindowsServer japanwest /Subscriptions...
2012-Datcenter WindowsServer MicrosoftWindowsServer japanwest /Subscriptions...
2012-R2-Datcenter WindowsServer MicrosoftWindowsServer japanwest /Subscriptions...
2016-Technical-Preview-3-with-Containers WindowsServer MicrosoftWindowsServer japanwest /Subscriptions...
2016-Technical-Preview-4-Nano-Server WindowsServer MicrosoftWindowsServer japanwest /Subscriptions...
2016-Technical-Preview-Nano-Server WindowsServer MicrosoftWindowsServer japanwest /Subscriptions...
Windows-Server-Technical-Preview WindowsServer MicrosoftWindowsServer japanwest /Subscriptions...
```

PublisherName が「MicrosoftSQLServer」で、Offer が「SQL2014SP1-WS2012R2」の仮想マシンイメージで選択できるすべての SKU を表示するには、以下のコマンドを実行します。

```
Get-AzureRmVMImageSkus -Location $location -Offer SQL2014SP1-WS2012R2 `
-PublisherName MicrosoftSQLServer
```

```
Get-AzureRmVMImageSkus -Location $location -Offer SQL2014SP1-WS2012R2 `
-PublisherName MicrosoftSQLServer

Skus          Offer          PublisherName      Location  Id
-----
Enterprise    SQL2014SP1-WS2012R2 MicrosoftSQLServer japanwest /Subscriptions/25443fab...
Enterprise-Optimized-for-DW SQL2014SP1-WS2012R2 MicrosoftSQLServer japanwest /Subscriptions/25443fab...
Enterprise-Optimized-for-QLTP SQL2014SP1-WS2012R2 MicrosoftSQLServer japanwest /Subscriptions/25443fab...
Standard      SQL2014SP1-WS2012R2 MicrosoftSQLServer japanwest /Subscriptions/25443fab...
Web           SQL2014SP1-WS2012R2 MicrosoftSQLServer japanwest /Subscriptions/25443fab...
```

Version には、「latest」を設定することで、最新の修正プログラムが適用されたイメージを使用することができます。

Set-AzureRmVMOSDisk に設定する主なパラメーターは下表のとおりです。

パラメーター	説明	必須
<b>Name</b>	OS ディスクの名前を指定します。	○
<b>VhdUri</b>	作成する VHD ファイルの URI パスを指定します。	
<b>Caching</b>	ディスク キャッシュ オプションを指定します。ReadOnly、ReadWrite、None から選択できます。OS ディスクの既定値は、ReadWrite です。	
<b>CreateOption</b>	OS ディスクの作成方法を指定します。Attach、Empty、FromImage から選択します。Azure のイメージ ギャラリーからプロビジョニングする場合、FromImage を指定します。	○

<b>VM</b>	設定する仮想マシン構成オブジェクトを指定します。	○
<b>SourceImageUri</b>	プロビジョニングで使用する OS イメージの URI パスを指定します。	

Add-AzureRmVMNetworkInterface に設定するパラメーターはすべて必須で下表のとおりです。

パラメーター	説明	必須
<b>id</b>	仮想マシンで使用するネットワーク インターフェイス リソースの ID を Get-AzureRmNetworkInterface で取得します。	○
<b>VM</b>	ネットワーク インターフェイスを設定する仮想マシン構成オブジェクトを指定します。	○

仮想マシンで使用する ネットワーク インターフェイスの ID を取得する Get-AzureRmNetworkInterface に必要な主なパラメーターは下表のとおりです。

パラメーター	説明	必須
<b>Name</b>	ネットワーク インターフェイスの名前を指定します。	
<b>ResourceGroupName</b>	ネットワーク インターフェイス作成されているリソース グループ名を指定します。	

自習書では、Web サーバーから作成します。New-AzureRmVM で、下表のパラメーターと変数値を使用します。

パラメーター	変数名	値
<b>ResourceGroupName</b>	\$rg	AW-RG
<b>Location</b>	\$location	Japanwest
<b>VM</b>	\$Config	New-AzureRmVMConfig で取得

New-AzureRmVM の VM パラメーターに必要な VM 構成オブジェクトは、下表のパラメーターと変数値を使用して、New-AzureRmVMConfig で作成します。

パラメーター	変数名	値
<b>VMName</b>	\$vmName	Web01
<b>VMSize</b>	\$vmSize	Standard_A1

VM 構成オブジェクトに、コンピューター名や仮想マシンに接続するための資格情報を設定する Set-AzureRMVMOperatingSystem で使用するパラメーターと値は下表のとおりです。

パラメーター	変数名	値
<b>ComputerName</b>	\$vmName	Web01
<b>Credential</b>	\$cred	Get-Credential で取得します。
<b>VM</b>	\$config	New-AzureRmVMConfig で作成します。
<b>Windows</b>	なし	なし
<b>ProvisionVMAgent</b>	なし	なし
<b>EnableAutoUpdate</b>	なし	なし

仮想マシンのソース イメージを指定する Set-AzureRmVMSourceImage で使用するパラメーターと値は下表のとおりです。

パラメーター	変数名	値
<b>PublisherName</b>	\$publisher	MicrosoftWindowsServer
<b>Offer</b>	\$offer	WindowsServer
<b>Skus</b>	\$sku	2012-R2-Datacenter
<b>Version</b>	\$version	Latest
<b>VM</b>	\$config	New-AzureRmVMConfig で作成します。

OS ディスクの配置先として使用する Blob ストレージのパスを設定する Set-AzureRmVMOSDisk で使用するパラメーターと値は下表のとおりです。

パラメーター	変数名	値
<b>Name</b>	\$vmName	Web01
<b>VhdUri</b>	\$vhdUri	https://\$StorageAccountName.blob.core.windows.net/vhds/\$vmName-os.vhd
<b>Caching</b>	\$caching	ReadWrite
<b>CreateOption</b>	\$createOption	fromImage
<b>VM</b>	\$config	New-AzureRmVMConfig で作成します。

ネットワーク インターフェイス リソースを仮想マシンに接続する Add-AzureRmVMNetworkInterface で使用するパラメーターと値は下表のとおりです。

パラメーター	変数名	値
<b>Id</b>	\$nic.Id	Get-AzureRmNetworkInterface で ネットワーク インターフェイスの情報を取得します。
<b>VM</b>	\$config	New-AzureRmVMConfig で作成します。

仮想マシンに接続する ネットワーク インターフェイスのリソース ID を取得する Get-AzureRmNetworkInterface で使用するパラメーターと値は下表のとおりです。

パラメーター	変数名	値
<b>Name</b>	\$nicName	Web01-NIC
<b>ResourceGroupName</b>	\$rg	AW-RG

1. [表示] メニューの [スクリプト ウィンドウの表示] を選択して、新しいスクリプト ウィンドウを開きます。
2. New-AzureRmVM のパラメーターに設定する変数を記述します。

```
# リソースグループ
$rg = "AW-RG"
# 地域
$location = "japanwest"
```



3. 変数宣言に New-AzureRmVMConfig のパラメーターに設定する変数を追加します。

```
# 仮想マシンの構成情報
$vmName = "Web01"
$vmSize = "Standard_A1"
```

4. 変数宣言に Set-AzureRmVMSourceImage のパラメーターに設定する変数を追加します。

```
# 仮想マシン イメージの情報
$publisher = "MicrosoftWindowsServer"
$offer = "WindowsServer"
$sku = "2012-R2-Datacenter"
$version = "latest"
```

5. 変数宣言に Set-AzureRmVMOSDisk のパラメーターに設定する変数を追加します。

```
# OS ディスクの情報
$StorageAccountName = "awvmstorageacct"
$caching = "ReadWrite"
$createOption = "fromImage"
$vhdUri = "https://$StorageAccountName.blob.core.windows.net/vhds/$vmName-os.vhd"
```

6. 変数宣言に Get-AzureRmNetworkInterface のパラメーターに設定する変数を追加します。

```
# NIC の情報
$nicName = "Web01-NIC"
```

7. 改行を入力して、仮想マシンで使用するネットワーク インターフェイス リソースの情報を取得するため、Get-AzureRmNetworkInterface を追加します。

```
# NIC の情報を取得
$nic = Get-AzureRmNetworkInterface -Name $nicName -ResourceGroupName $rg
```

8. 仮想マシンの管理者アカウントを保存するため Get-Credential を追加します。

```
# 仮想マシンの管理者アカウント情報を取得
$cred = Get-Credential
```

9. 仮想マシンの構成オブジェクトを作成するための New-AzureRmVMConfig コマンドレットと、構成情報を設定するための「Set-」および「Add-」で始まる一連のコマンドレットをパイプで接続して記述します。

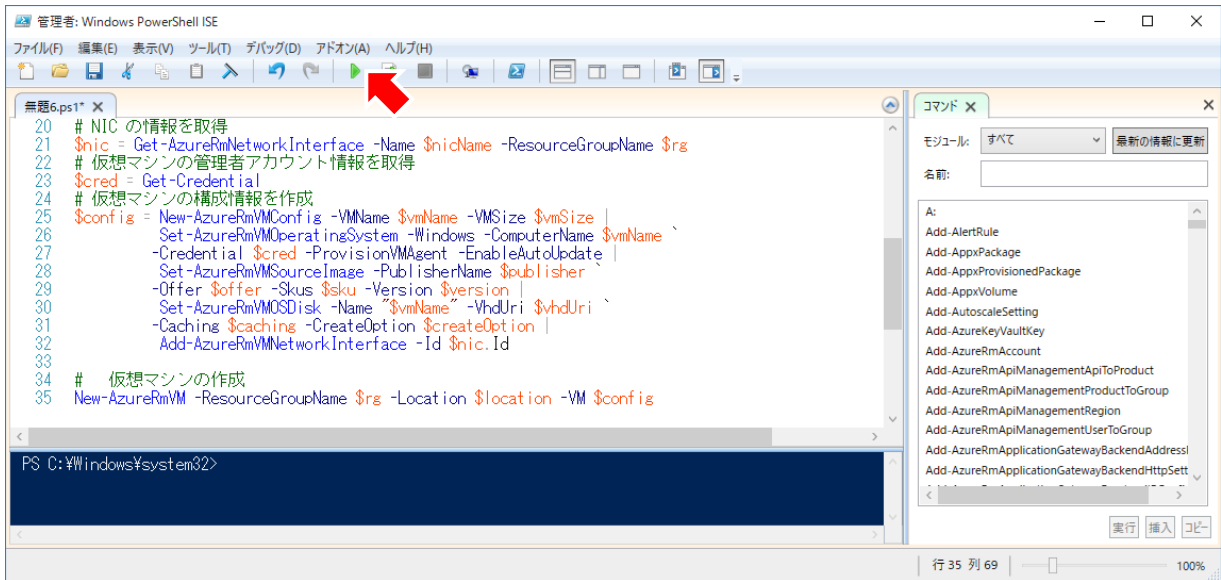
```
# 仮想マシンの構成情報を作成
$config = New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize |
  Set-AzureRmVMOperatingSystem -Windows -ComputerName $vmName `
  -Credential $cred -ProvisionVMAgent -EnableAutoUpdate |
  Set-AzureRmVMSourceImage -PublisherName $publisher `
  -Offer $offer -Skus $sku -Version $version |
  Set-AzureRmVMOSDisk -Name "$vmName" -VhdUri $vhdUri `
  -Caching $caching -CreateOption $createOption |
  Add-AzureRmVMNetworkInterface -Id $nic.Id
```

10. 作成した仮想マシンの構成オブジェクトを使用して、仮想マシンを作成するための New-AzureRmVM コマンドレットを記述します。

```
# 仮想マシンの作成

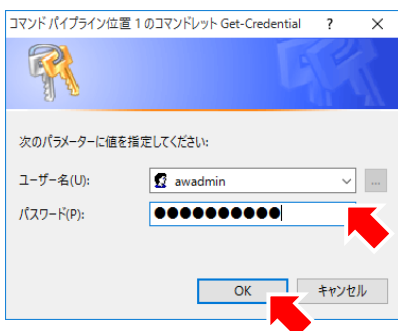
New-AzureRmVM -ResourceGroupName $rg -Location $location -VM $config
```

11. Windows PowerShell ISE のスクリプト ウィンドウで [スクリプトを実行] ボタンをクリックして、作成したスクリプトを実行します。



12. Get-Credential より資格情報の入力を要求するダイアログボックスが表示されるので下表の内容を入力して [OK] ボタンをクリックします。

パラメーター	説明	設定する値
ユーザー名	管理者のユーザー名です。	awadmin
パスワード	管理者のパスワードです。	awPa\$\$w0rd



### ワンポイント

この後、仮想マシンのプロビジョニングが開始します。仮想マシンが作成されインスタンスが開始するまで、10 分程度かかります。進捗状況は、Get-AzureRmVM コマンドレットで、ProvisioningState を表示することで確認することができます。

```
Get-AzureRmVM -Name $vmName -ResourceGroupName $rg | `
select Name, ProvisioningState
```

```
Get-AzureRmVM -Name $vmName -ResourceGroupName $rg | `
select Name, ProvisioningState

Name ProvisioningState
-----
Web01 Creating
```

パブリック IP アドレスが設定されたことを確認するには、Get-AzureRmPublicIpAddress コマンドレットを使用します。

```
Get-AzureRmPublicIpAddress -ResourceGroupName $rg | `
select Name,IpAddress
```

```
Get-AzureRmPublicIpAddress -ResourceGroupName $rg | `
select Name,IpAddress

Name      IpAddress
-----
SQL01-IP Not Assigned
Web01-IP 40.74.132.37
```

インスタンスが起動している場合、パブリック IP アドレスが割り当てられたことが確認できます。

13. New-AzureRmVM の出力で Status が「Succeeded」で StatusCode が「OK」が返されれば仮想マシンの作成は完了です。

```
Status           : Succeeded
StatusCode       : OK
RequestId        : 39ab1565-66c0-4c4c-9731-c700ebab5f32
Output           :
Error            :
StartTime        : 2015/11/11 14:06:35 +09:00
EndTime          : 2015/11/11 14:12:53 +09:00
TrackingOperationId : 52245a06-9359-475f-8d60-68f339709c99
```

14. Windows PowerShell ISE の [ファイル] メニューから [名前を付けて保存] を選択し、作成したスクリプトを作業用フォルダーに、「**Create-VM-for-Web01.ps1**」という名前で保存します。作成したスクリプトは、以下のコードになります。

```
# リソースグループ
$rg = "AW-RG"
# 地域
$location = "japanwest"
# 仮想マシンの構成情報
$vmName = "Web01"
$vmSize = "Standard_A1"
# 仮想マシン イメージの情報
$publisher = "MicrosoftWindowsServer"
$offer = "WindowsServer"
$sku = "2012-R2-Datacenter"
$version = "latest"
# OS ディスクの情報
$StorageAccountName = "awvmstorage"
$caching = "ReadWrite"
$createOption = "fromImage"
$vhdUri = "https://$StorageAccountName.blob.core.windows.net/vhds/$vmName-os.vhd"
```

```
# NIC の情報
$nicName = "Web01-NIC"

# NIC の情報を取得
$nic = Get-AzureRmNetworkInterface -Name $nicName -ResourceGroupName $rg
# 仮想マシンの管理者アカウント情報を取得
$cred = Get-Credential
# 仮想マシンの構成情報を作成
$config = New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize |
    Set-AzureRmVMOperatingSystem -Windows -ComputerName $vmName `
    -Credential $cred -ProvisionVMAgent -EnableAutoUpdate |
    Set-AzureRmVMSourceImage -PublisherName $publisher `
    -Offer $offer -Skus $sku -Version $version |
    Set-AzureRmVMOSDisk -Name "$vmName" -VhdUri $vhdUri `
    -Caching $caching -CreateOption $createOption |
    Add-AzureRmVMNetworkInterface -Id $nic.Id
# 仮想マシンの作成
New-AzureRmVM -ResourceGroupName $rg -Location $location -VM $config
```

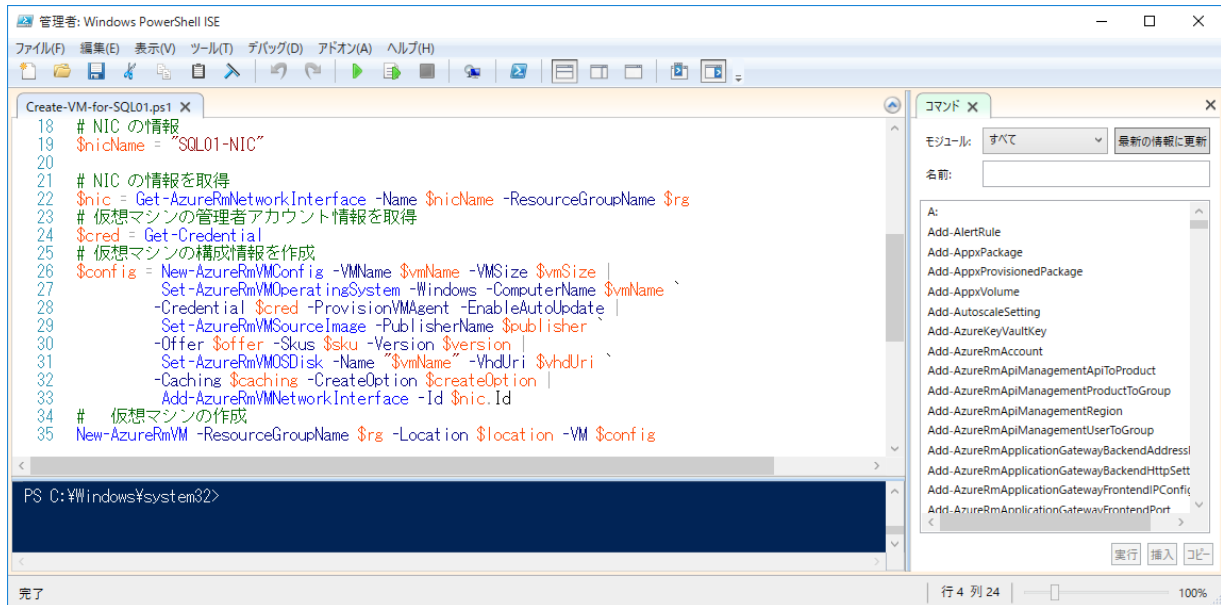
15. 次に、データベース サーバー用の仮想マシンを作成するために、下表を参考にして、変数宣言の初期値を変更して、[ファイル] メニューから [名前を付けて保存] を選択し、作成したスクリプトを作業用フォルダーに、「**Create-VM-for-SQL01.ps1**」という名前で保存します。

変数名	説明	Web サーバーの設定	DB サーバーの設定
\$vmName	仮想マシンの名前	Web01	SQL01
\$publisher	仮想マシン イメージのパブリッシャー名	MicrosoftWindowsServer	MicrosoftSQLServer
\$offer	仮想マシン イメージのオファー名	WindowsServer	SQL2014SP1-WS2012R2
\$sku	仮想マシン イメージの SKU	2012-R2-Datacenter	Enterprise
\$nicName	ネットワーク インターフェイスの名前	Web01-NIC	SQL01-NIC

```
# リソースグループ
$rg = "AW-RG"
# 地域
$location = "japanwest"
# 仮想マシンの構成情報
$vmName = "SQL01"
$vmSize = "Standard_A1"
# 仮想マシン イメージの情報
$publisher = "MicrosoftSQLServer"
$offer = "SQL2014SP1-WS2012R2"
$sku = "Enterprise"
$version = "latest"
# OS ディスクの情報
$StorageAccountName = "awvmstorageacct"
$caching = "ReadWrite"
$createOption = "fromImage"
```

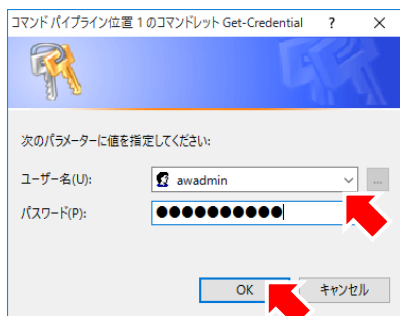
```
$vhdUri = "https://$StorageAccountName.blob.core.windows.net/vhds/$vmName-os.vhd"
# NIC の情報
$nicName = "SQL01-NIC"
```

16. Windows PowerShell ISE のスクリプト ウィンドウで [スクリプトを実行] ボタンをクリックして、作成したスクリプトを実行します。



17. Get-Credential より資格情報の入力を要求するダイアログボックスが表示されるので下表の内容を入力して [OK] ボタンをクリックします。

パラメーター	説明	設定する値
ユーザー名	管理者のユーザー名です。	awadmin
パスワード	管理者のパスワードです。	awPa\$\$w0rd



### ワンポイント

この後、仮想マシンのプロビジョニングが開始します。仮想マシンが作成されインスタンスが開始するまで、10 分程度かかります。進捗状況は、Get-AzureRmVM コマンドレットで、ProvisioningState を表示することで確認することができます。

```
Get-AzureRmVM -Name $vmName -ResourceGroupName $rg | `
select Name, ProvisioningState
```

```
Get-AzureRmVM -Name $vmName -ResourceGroupName $rg | `
select Name, ProvisioningState

Name ProvisioningState
-----
SQL01 Creating
```

パブリック IP アドレスが設定されたことを確認するには、Get-AzureRmPublicIpAddress コマンドレットを使用します。

```
Get-AzureRmPublicIpAddress -ResourceGroupName $rg | `
select Name,IpAddress
```

```
Get-AzureRmPublicIpAddress -ResourceGroupName $rg | `
select Name,IpAddress

Name      IpAddress
-----
SQL01-IP  40.74.124.211
Web01-IP  40.74.132.37
```

インスタンスが起動している場合、パブリック IP アドレスが割り当てられたことが確認できます。

18. New-AzureRmVM の出力で Status が「Succeeded」で StatusCode が「OK」が返されれば仮想マシンの作成は完了です。

```
Status           : Succeeded
StatusCode        : OK
RequestId         : 36533e26-4b1b-4727-afc1-0dc3b743f73f
Output            :
Error             :
StartTime         : 2015/11/11 14:40:12 +09:00
EndTime          : 2015/11/11 14:58:10 +09:00
TrackingOperationId : 9a135cb8-e968-43a3-be01-346e546ec211
```

19. 作成したスクリプトは、以下のコードになります。

```
# リソースグループ
$rg = "AW-RG"
# 地域
$location = "japanwest"
# 仮想マシンの構成情報
$vmName = "SQL01"
$vmSize = "Standard_A1"
# 仮想マシン イメージの情報
$publisher = "MicrosoftSQLServer"
$offer = "SQL2014SP1-WS2012R2"
$sku = "Enterprise"
$version = "latest"
# OS ディスクの情報
$StorageAccountName = "awvmstorage"
$caching = "ReadWrite"
$createOption = "fromImage"
$vhdUri = "https://$StorageAccountName.blob.core.windows.net/vhds/$vmName-os.vhd"
# NIC の情報
$nicName = "SQL01-NIC"
```

```
# NIC の情報を取得

$nic = Get-AzureRmNetworkInterface -Name $nicName -ResourceGroupName $rg
# 仮想マシンの管理者アカウント情報を取得

$cred = Get-Credential
# 仮想マシンの構成情報を作成

$config = New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize |
    Set-AzureRmVMOperatingSystem -Windows -ComputerName $vmName `
    -Credential $cred -ProvisionVMAgent -EnableAutoUpdate |
    Set-AzureRmVMSourceImage -PublisherName $publisher `
    -Offer $offer -Skus $sku -Version $version |
    Set-AzureRmVMOSDisk -Name "$vmName" -VhdUri $vhdUri `
    -Caching $caching -CreateOption $createOption |
    Add-AzureRmVMNetworkInterface -Id $nic.Id

# 仮想マシンの作成

New-AzureRmVM -ResourceGroupName $rg -Location $location -VM $config
```

### ワンポイント

ここまでの手順で作成したリソースを確認するには、Get-AzureRmResource コマンドレットを使用します。

```
Get-AzureRmResource | ? { $_.ResourceGroupName -eq $rg } | Select Name, ResourceType
```

```
PS C:\Windows\system32> Get-AzureRmResource -ResourceGroupName $rg | Select Name, ResourceType

Name                ResourceType
----                -
awvmstorage         Microsoft.Storage/storageAccounts
AW-VNet             Microsoft.Network/virtualNetworks
SQL01               Microsoft.Compute/virtualMachines
SQL01-IP            Microsoft.Network/publicIPAddresses
SQL01-NIC           Microsoft.Network/networkInterfaces
Web01               Microsoft.Compute/virtualMachines
Web01/Web01         Microsoft.Compute/virtualMachines/extensions
Web01-IP            Microsoft.Network/publicIPAddresses
Web01-NIC           Microsoft.Network/networkInterfaces
```

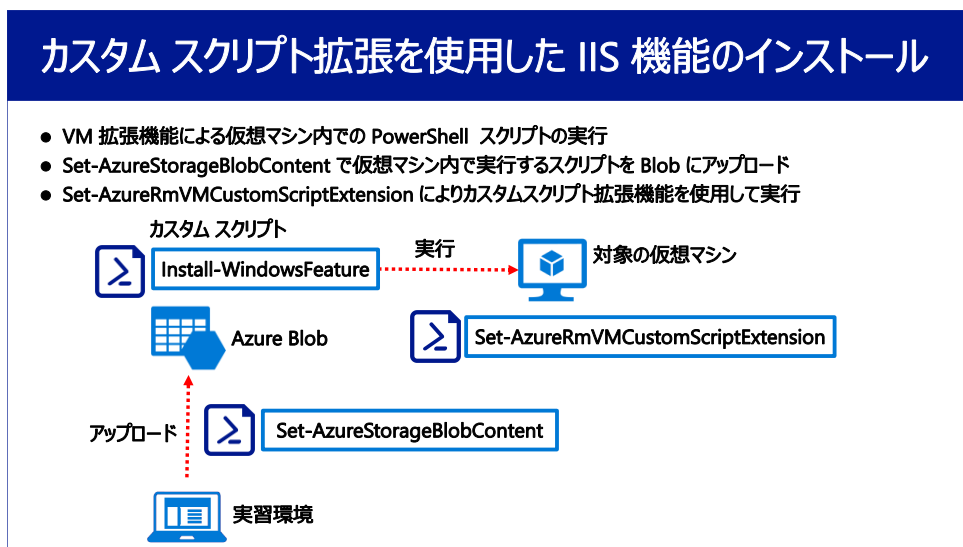
作成した 仮想マシンを削除するには、Remove-AzureRmVM コマンドレットを使用します。

```
Remove-AzureRmVM -Name $vmName -ResourceGroupName $rg
```

13. [Create-VM-for-SQL01.ps1] タブを閉じます。

### 3.6 カスタム スクリプト拡張を使用した IIS 機能のインストール

Azure PowerShell コマンドレットで Azure 仮想マシンのカスタム スクリプト拡張機能を使用する方法を説明します。カスタム スクリプト拡張機能を使うと、仮想マシン インスタンスに対して、リモートから PowerShell スクリプトを実行できます。スクリプトは、仮想マシンのプロビジョニング後、インスタンスの稼働中であればいつでも実行できます。仮想マシンにログインする必要も、新たにポートを開く必要もありません。この機能は、OS のプロビジョニング後、追加のソフトウェアをインストールし、構成するようなシナリオで使用します。



本書では、前の手順で作成した Web サーバーのための仮想マシンインスタンスに対して、カスタム スクリプトを実行し IIS をインストールします。カスタム スクリプトは、ローカル コンピューターで作成し、Azure Blob ストレージにアップロードした後、Set-AzureRmVMCustomScriptExtension コマンドレットを実行して、対象の仮想マシン インスタンスにダウンロードして実行します。Set-AzureRmVMCustomScriptExtension を使用すると 仮想マシンのカスタム スクリプト拡張機能により、インスタンスの中で、PowerShell スクリプトを実行することができます。

1. IIS をインストールするカスタム スクリプトを作成するため [ファイル] メニューから [新規作成] を選択して、新しいタブを開き、以下のコードを入力します。

```
Install-WindowsFeature Web-Server -IncludeAllSubFeature
```

2. [ファイル] メニューから [名前を付けて保存] を選択し、作成したスクリプトを作業用フォルダーに、「**Install-IIS.ps1**」という名前で保存します。
3. [Install-IIS.ps1] タブを閉じます。
4. 作成したスクリプトを Blob ストレージに保存するため、Windows PowerShell ISE の [ファイル] メニューから [新規作成] を選択して、新しいタブを開き、以下のコードを入力して、[スクリプトを実行] ボタンをクリックして、実行します。

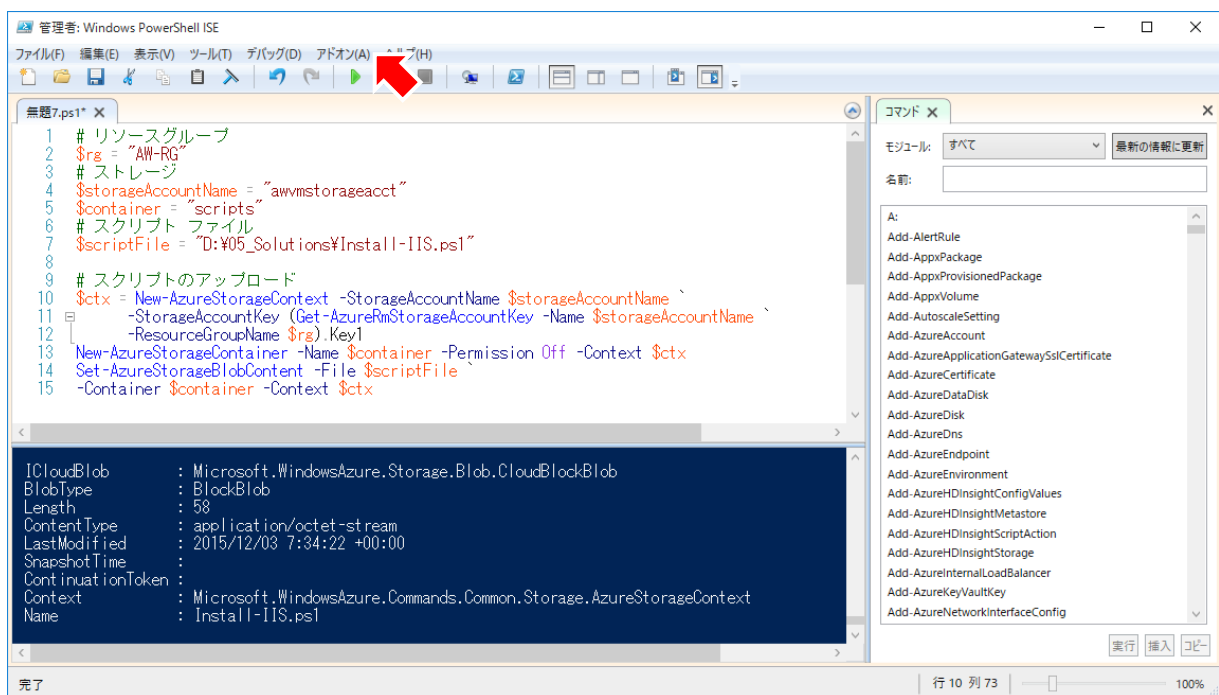
```
# リソースグループ
$rg = "AW-RG"
# ストレージ
```



```

$storageAccountName = "awvmstorageacct"
$container = "scripts"
# スクリプト ファイル
$scriptFile = "D:¥Solutions¥Install-IIS.ps1"

# スクリプトのアップロード
$ctx = New-AzureStorageContext -StorageAccountName $storageAccountName `
    -StorageAccountKey (Get-AzureRmStorageAccountKey -Name $storageAccountName `
    -ResourceGroupName $rg).Key1
New-AzureStorageContainer -Name $container -Permission Off -Context $ctx
Set-AzureStorageBlobContent -File $scriptFile `
    -Container $container -Context $ctx
    
```



### ワンポイント

このスクリプトにより、指定したストレージ アカウントに、「scripts」という名前でコンテナが作成され、前の手順で作成した「Install-IIS.ps1」が Blob ストレージにアップロードされます。このスクリプトでは、「Install-IIS.ps1」は、D ドライブの Solutions フォルダーに置かれているものとしています。ファイル パスは、ご自身の実習環境に合うように修正してください。

5. [ファイル] メニューから [名前を付けて保存] を選択し、作成したスクリプトを作業用フォルダーに、「**Upload-Script.ps1**」という名前で保存します。
6. [Upload-Script.ps1] タブを閉じます。
7. [表示] メニューの [スクリプト ウィンドウの表示] を選択して、新しいスクリプト ウィンドウを開きます。

8. 変宣言のための以下のコードを記述します。

```
# リソースグループ

$rg = "AW-RG"

# 仮想マシンの情報

$vmName = "Web01"

# ストレージ

$storageAccountName = "awvmstorageacct"

# カスタムスクリプト ファイルの情報

$container = "scripts"
$scriptFile = "Install-IIS.ps1"
```

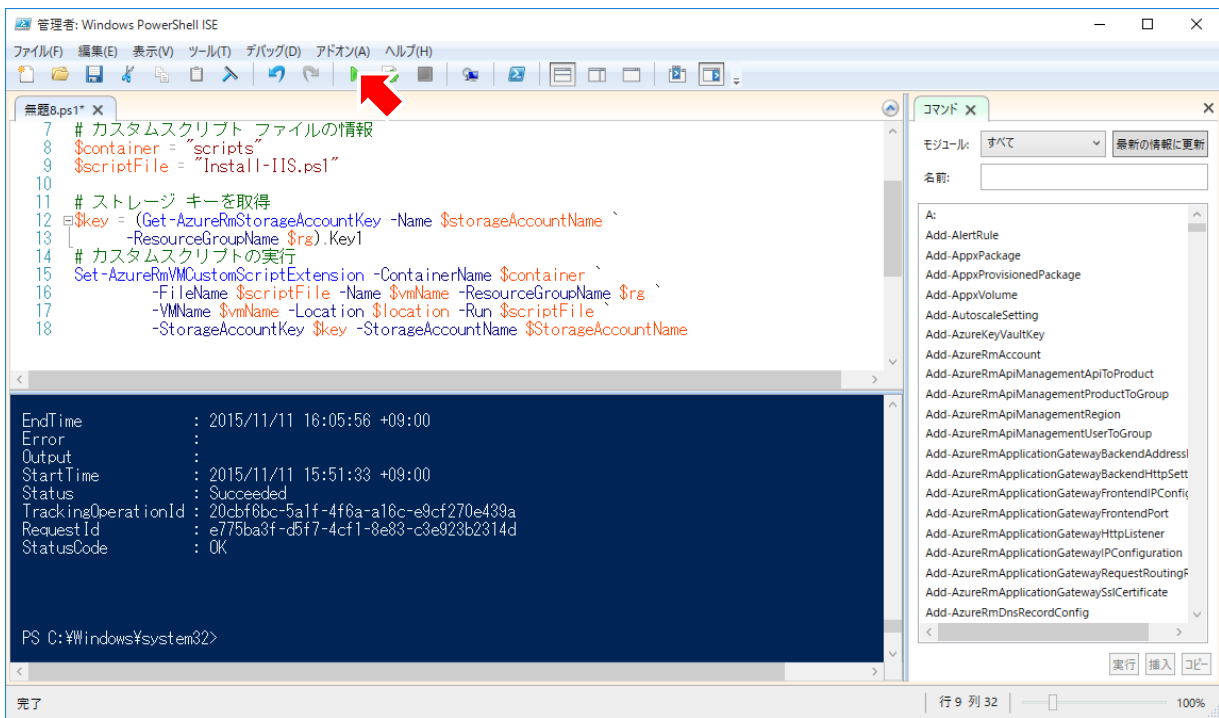
9. 改行を入力した後、以下のコードを入力して、[スクリプトを実行] ボタンをクリックして、実行します。

```
# ストレージ キーを取得

$key = (Get-AzureRmStorageAccountKey -Name $storageAccountName `
    -ResourceGroupName $rg).Key1

# カスタムスクリプトの実行

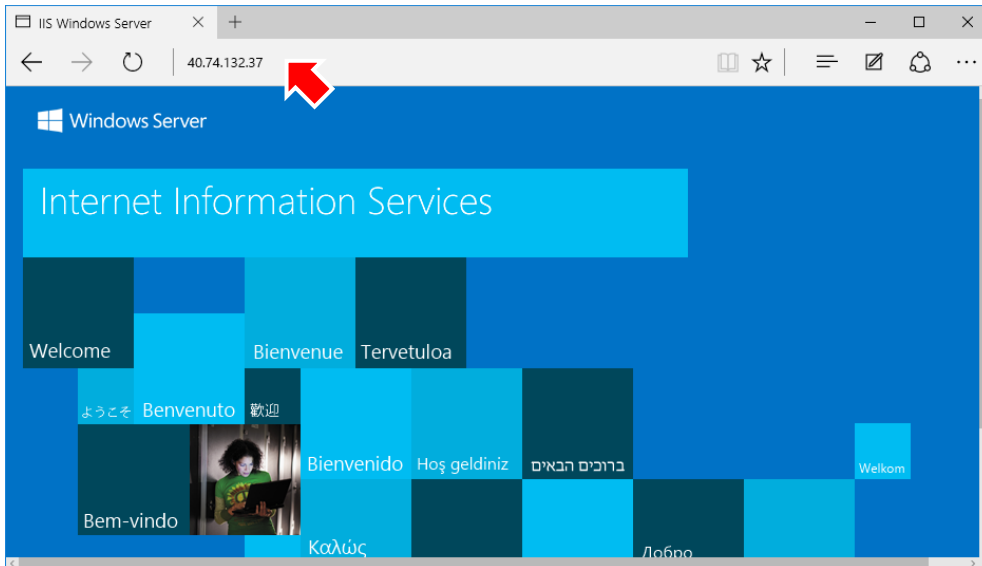
Set-AzureRmVMCustomScriptExtension -ContainerName $container `
    -FileName $scriptFile -Name $vmName -ResourceGroupName $rg `
    -VMName $vmName -Location $location -Run $scriptFile `
    -StorageAccountKey $key -StorageAccountName $StorageAccountName
```



**ワンポイント**  
 このスクリプトの実行には、10 分程度の時間がかかります。このスクリプトにより、Blob ストレージに配置したカスタム スクリプトが、仮想マシンで実行され、IIS がインストールされます。

本書では、Set-AzureRmVMCustomScriptExtension を使用して IIS をインストールする方法を説明しましたが、PowerShell DSC (Desired State Configuration) を使用して IIS をインストールする場合は、Set-AzureRmVMDscExtension を使用してください。また、JSON テンプレートを使用した PowerShell DSC 拡張を使用する IIS のインストール手順は、「4.6 DSC 拡張機能を使用した IIS のインストール」で確認してください。

10. Web ブラウザーを起動して、アドレス バーに、「**http://<Web01 のパブリック IP アドレス>**」と入力して IIS の既定のページが参照できることを確認します。



#### ワンポイント

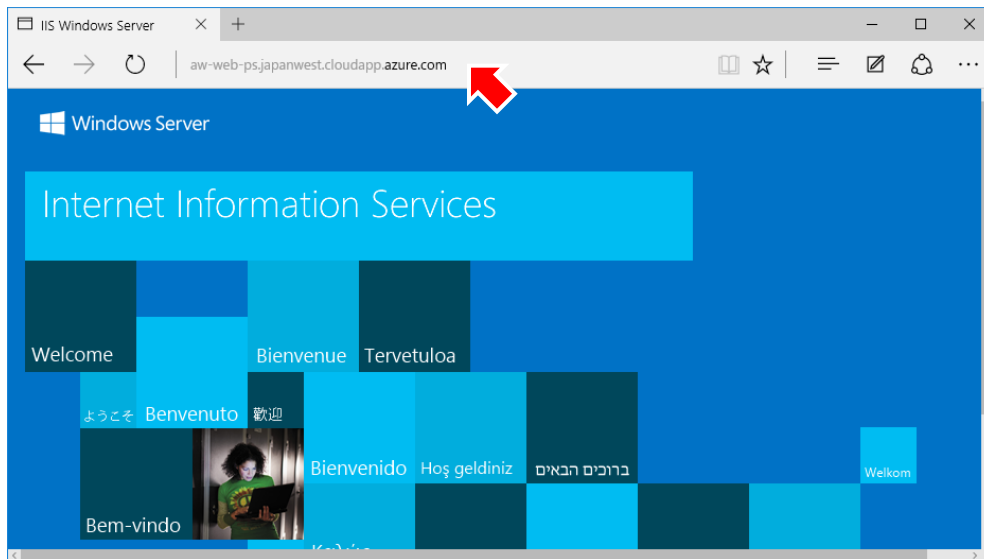
パブリック IP アドレスの確認は、Get-AzureRmPublicIpAddress コマンドレットを使用します。

```
Get-AzureRmPublicIpAddress -ResourceGroupName $rg | `
select Name,IpAddress
```

```
Get-AzureRmPublicIpAddress -ResourceGroupName $rg | `
select Name,IpAddress

Name      IpAddress
-----
SQL01-IP  40.74.124.211
Web01-IP  40.74.132.37
```

- 次に、パブリック IP アドレス リソースに割り当てた ドメイン名ラベルを使用して IIS にアクセスするために Web ブラウザーのアドレス バーに、「<http://aw-web-ps.japanwest.cloudapp.azure.com>」と入力して既定のページが参照できることを確認します。



#### ワンポイント

ドメイン名ラベルの「aw-web-ps」は、「3.4 ネットワーク インターフェイスとパブリック IP アドレスの作成」でご自身が設定したラベルに変更して実行してください。

- [ファイル] メニューから [名前を付けて保存] を選択し、作成したスクリプトを作業用フォルダーに、「**Run-CustomScript.ps1**」という名前で保存します。
- [Run-CustomScript.ps1] タブを閉じます。

### 3.7 作成したリソース グループの削除

ここでは、作成したリソース グループを削除します。個々のリソースを個別に削除することもできますが、リソース グループを削除することで、そのリソース グループ内のすべてのリソースを削除することができます。AW-RG リソース グループを削除することで、その中にあるリソースをすべてが削除することができます。

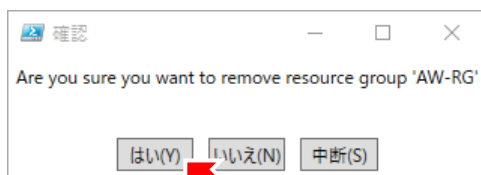
なお、次の 4 章の実習では、3 章で作成したリソースと同じ名前のリソースを JSON テンプレート ファイルを読み込んで作成します。そのため、以降の手順を行う場合、事前に、ここまで作成したリソースの削除しておいてください。

1. Windows PowerShell ISE の [表示] メニューの [スクリプト ウィンドウの表示] を選択して、新しいスクリプト ウィンドウを開きます。
2. スクリプト ウィンドウに以下のコードを記述した後、[スクリプトを実行] ボタンをクリックして、実行します。

```
# リソースグループ
$rg = "AW-RG"

# リソースグループの削除
Remove-AzureRmResourceGroup -Name $rg
```

3. 以下の確認メッセージが表示されたら [はい] ボタンをクリックします。



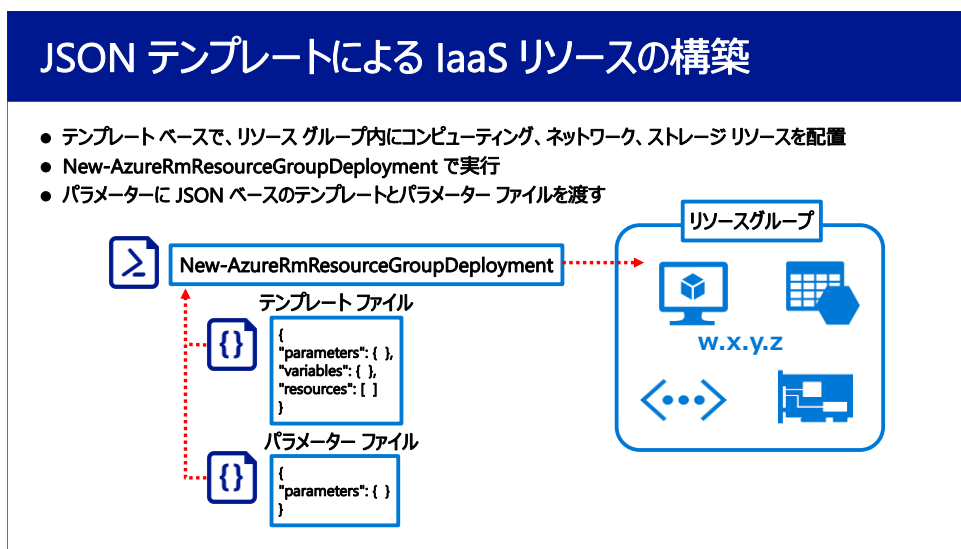
4. リソースグループの削除が完了したら、以下のコードを入力し、選択状態にして、[選択項目を実行] ボタンで実行します。表示される一覧にリソースグループ「AW-RG」が存在しないことを確認します。

```
# リソースグループの確認
Get-AzureRmResourceGroup | Select ResourceGroupName
```

5. Windows PowerShell ISE の [ファイル] メニューから [名前を付けて保存] を選択し、作成したスクリプトを作業用フォルダーに、「**Remove-ResourceGroup.ps1**」という名前で保存します。
6. [Remove-ResourceGroup.ps1] タブを閉じます。

## 4. PowerShell と JSON テンプレートによる IaaS v2 リソースの構築

3 章では、IaaS v2 のリソースを作成するための各コマンドレットを確認しながら実行し、Web サーバーとデータベースサーバーのための仮想マシン環境を構築してきました。Azure PowerShell では、New-AzureRmResourceGroupDeployment コマンドレットを使用することで、Azure に配置する必要があるリソースを宣言的に定義したテンプレート ファイルを読み込んで、作成することもできます。また、この方法のよい点としては、構成を変更する場合、New- 系のコマンドレットで作成し、Set- 系で変更といったように、やりたいことに応じたコマンドの選択を考え、スクリプトを実行するのではなく、テンプレート ファイルの定義を変更して、再度、New-AzureRmResourceGroupDeployment コマンドレットを実行するだけで済むことです。



テンプレートは、データ記述言語の 1 つである JSON (JavaScript Object Notation) 表記で定義します。JSON では、オブジェクトの始まりと終わりを中括弧「{ }」で囲んで表現し、名前 (キー) と値がペアになった複数の要素をカンマで区切って記述します。また、リソースの配置で使用する値には、関数や式を含めることができます。テンプレートの基本構造は以下のとおりで、太字で示した、`$schema`、`contentVersion`、`resources` は必須です。

```
{
  "$schema": "http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion": "",
  "parameters": { },
  "variables": { },
  "resources": [ ],
  "outputs": { }
}
```

この中でも `resources` の定義が重要で、ここに配置する各リソースの内容を記載します。variables は、変数宣言のために使用され、parameters は、プロビジョニング時に使用されるパラメーター値を定義します。以降で、これらの要素を定義したテンプレート ファイルを作成し、IaaS v2 リソースを配置する方法を確認します。

4 章の実習は、3 章で作成したリソースと同じ名前のリソースを JSON テンプレート ファイルを読み込んで作成します。

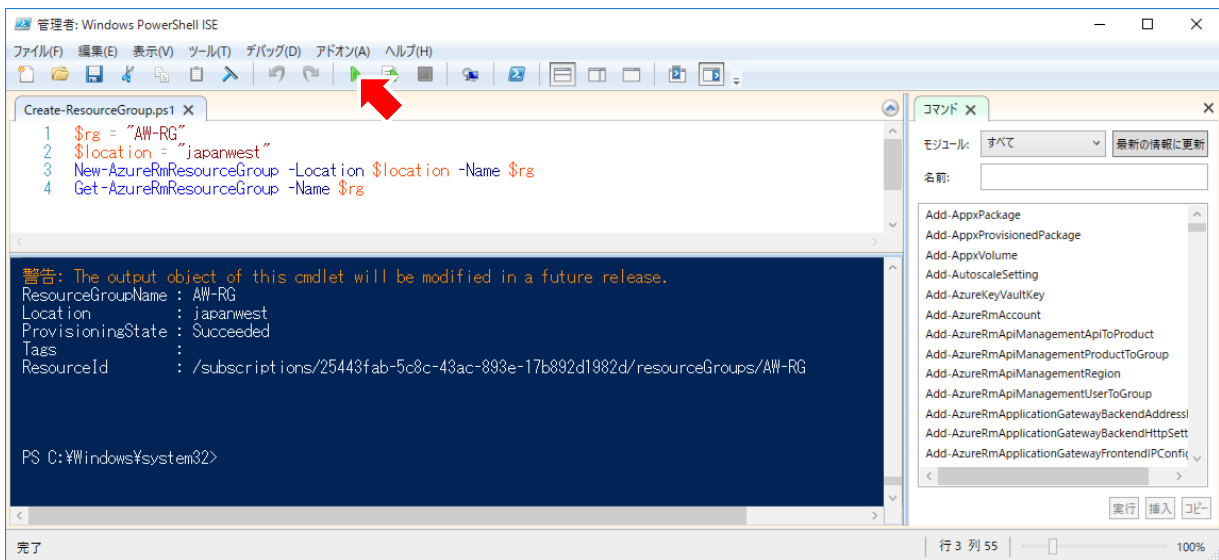
そのため、この章の手順を行う場合は、事前に、「3.8 作成したリソース グループの削除」を行うようにしてください。また、JSON テンプレートで定義できるリソースに、リソースグループは含まれていないため、まず、事前に、New-AzureRmResourceGroup コマンドレットで、リソース グループを作成しておく必要があります。

#### 4.1 リソース グループの作成

1. Windows PowerShell ISE の [ファイル] メニューから [開く] を選択し、3 章で作成した「Create-ResourceGroup.ps1」を開きます。スクリプトは、以下のコードになります。

```
$rg = "AW-RG"
$location = "japanwest"
New-AzureRmResourceGroup -Location $location -Name $rg
Get-AzureRmResourceGroup -Name $rg
```

2. [スクリプトを実行] ボタンをクリックして実行し、リソース グループを作成します。



3. [Create-ResourceGroup.ps1] タブを閉じます。

## 4.2 ストレージ アカウントの作成

最初に、オプションの要素として変数宣言のための `variables` を使用したシンプルなテンプレート ファイルを作成して、ストレージ アカウント (`Microsoft.Storage/storageAccounts`) を作成してみます。JSON を編集するエディターは、任意の製品をご利用いただけます。Microsoft 製品では、<https://www.visualstudio.com/ja-jp/downloads/download-visual-studio-vs.aspx> から無償でダウンロードすることができる Visual Studio 2015 Express for Web の JSON エディターを使用することができます。

1. JSON エディターを起動して、ストレージ アカウントを作成するための以下の JSON コードを記述し、作業用フォルダーに、「**deploy-storage.json**」という名前で保存します。

```
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "variables": {
    "location": "japanwest",
    "storageAccountName": "awvmstorageacct",
    "storageAccountType": "Standard_LRS"
  },
  "resources": [
    {
      "apiVersion": "2015-05-01-preview",
      "type": "Microsoft.Storage/storageAccounts",
      "name": "[variables('StorageAccountName')]",
      "location": "[variables('location')]",
      "properties": {
        "accountType": "[variables('storageAccountType')]"
      }
    }
  ]
}
```

### ワンポイント

本自習書では、ストレージ アカウント名として、「awsqldbstorageacct」を使用するように記載していますが、名前は、Azure で一意な名前を設定する必要があることに注意してください。実習を行う場合、「aw」の部分を別の文字列に置き換えるか任意の数値を付加するようにしてください。

### ワンポイント

リソース (`resources`) の定義から `variables` を参照する場合、`variables()` 関数を使用することができます。ここで掲載した JSON ファイルは、自習書のサイトから **Solutions.zip** ファイルをダウンロードすることで取得できます。ZIP ファイルには、自習書で作成するすべてのスクリプト ファイルと JSON ファイルが含まれています。

2. Windows PowerShell ISE の [表示] メニューの [スクリプト ウィンドウの表示] を選択して、新しいスクリプト ウィンドウを開きます。



3. 保存したテンプレート ファイルを使用して、ストレージアカウントを作成するため、次のコードを記述して、[スクリプトを実行] ボタンをクリックして実行します。

```
# リソース グループ名
$rg = "AW-RG"

# テンプレート ファイルを使用したストレージ アカウントの作成
New-AzureRmResourceGroupDeployment -ResourceGroupName $rg `
-TemplateFile D:¥Solutions¥deploy-storage.json
```

#### ワンポイント

このスクリプトは、JSON ファイルが D ドライブの Solutions フォルダーに保存されているものとしています。異なるフォルダーを使用している場合、適宜、フォルダー名を変更してください。

4. ストレージ アカウントの作成が完了すると次のメッセージが返されます。

```
DeploymentName      : deploy-storage
ResourceGroupName  : AW-RG
ProvisioningState  : Succeeded
Timestamp          : 2015/11/12 6:50:51
Mode               : Incremental
TemplateLink       :
Parameters         :
Outputs            :
```

#### ワンポイント

ProvisioningState が Succeeded になっていることを確認します。

5. 作成されたリソースを確認するため、次のコードを記述して、選択状態にした後、[選択項目を実行] ボタンで実行します。

```
# 作成されたリソースの確認
Get-AzureRmResource | ? {$_.ResourceGroupName -eq $rg } | Select Name, ResourceType
```

```
PS C:¥Windows¥system32> # 作成されたリソースの確認
Get-AzureRmResource | ? {$_.ResourceGroupName -eq $rg } | Select Name, ResourceType

Name                ResourceType
----                -
awvmstorageacct    Microsoft.Storage/storageAccounts
```

6. Windows PowerShell ISE の [ファイル] メニューから [名前を付けて保存] を選択し、作成したスクリプトを作業用フォルダーに、「**Create-StorageAccount-ByTemplate.ps1**」という名前で保存します。

### 4.3 仮想ネットワークとサブネットの作成

次に、テンプレート ファイルを使用して 2 つのサブネットを持つ仮想ネットワーク (Microsoft.Network/virtualNetworks) を作成します。

1. JSON エディターに、仮想ネットワークを作成するための以下の JSON コードを記述し、作業用フォルダーに、「**deploy-vnet.json**」という名前で保存します。

```
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "variables": {
    "location": "japanwest",
    "virtualNetworkName": "AW-VNet",
    "addressPrefix": "192.168.1.0/25",
    "subnet1Name": "Frontend-SN",
    "subnet1Prefix": "192.168.1.0/27",
    "subnet2Name": "Backend-SN",
    "subnet2Prefix": "192.168.1.32/27"
  },
  "resources": [
    {
      "apiVersion": "2015-05-01-preview",
      "type": "Microsoft.Network/virtualNetworks",
      "name": "[variables('virtualNetworkName')]",
      "location": "[variables('location')]",
      "properties": {
        "addressSpace": {
          "addressPrefixes": [
            "[variables('addressPrefix')]"
          ]
        },
        "subnets": [
          {
            "name": "[variables('subnet1Name')]",
            "properties": {
              "addressPrefix": "[variables('subnet1Prefix')]"
            }
          },
          {
            "name": "[variables('subnet2Name')]",
            "properties": {
              "addressPrefix": "[variables('subnet2Prefix')]"
            }
          }
        ]
      }
    }
  ]
}
```

```

    }
  ]
}

```

**ワンポイント**

この JSON ファイルは、自習書のサイトから **Solutions.zip** ファイルをダウンロードすることで取得できます。ZIP ファイルには、自習書で作成するすべてのスクリプト ファイルと JSON ファイルが含まれています。

- 保存したテンプレート ファイルを使用して、仮想ネットワークを作成するため Windows PowerShell ISE のスクリプトを次のコードに変更（以下の赤字部分）して、選択状態にした後、[選択項目を実行] ボタンで実行します。

```

# リソース グループ名
$rg = "AW-RG"

# テンプレート ファイルを使用した仮想ネットワークの作成
New-AzureRmResourceGroupDeployment -ResourceGroupName $rg `
-TemplateFile D:\%Solutions%\bdeploy-vnet.json

```

**ワンポイント**

このスクリプトは、JSON ファイルが D ドライブの Solutions フォルダに保存されているものとしています。異なるフォルダを使用している場合、適宜、フォルダ名を変更してください。

- 仮想ネットワークの作成が完了すると次のメッセージが返されます。

```

DeploymentName      : deploy-vnet
ResourceGroupName  : AW-RG
ProvisioningState   : Succeeded
Timestamp           : 2015/11/12 6:54:18
Mode                : Incremental
TemplateLink        :
Parameters          :
Outputs             :

```

**ワンポイント**

ProvisioningState が Succeeded になっていることを確認します。

- 作成されたリソースを確認するため、次のコードを選択状態にして、[選択項目を実行] ボタンで実行します。

```

# 作成されたリソースの確認
Get-AzureRmResource | ? {$_.ResourceGroupName -eq $rg } | Select Name, ResourceType

```

```

PS C:\Windows\system32> # 作成されたリソースの確認
Get-AzureRmResource | ? {$_.ResourceGroupName -eq $rg } | Select Name, ResourceType

Name                ResourceType
----                -
AW-VNet              Microsoft.Network/virtualNetworks
awvmstorageacct     Microsoft.Storage/storageAccounts

```

- Windows PowerShell ISE の [ファイル] メニューから [名前を付けて保存] を選択し、作成したスクリプトを作業用フォルダに、「**Create-VNet-ByTemplate.ps1**」という名前で保存します。

#### 4.4 ネットワーク インターフェイスとパブリック IP アドレスの作成

テンプレート ファイルを使用して仮想マシンで使用するためのネットワーク インターフェイス リソース (Microsoft.Network/virtualNetworks) と ネットワーク インターフェイス リソースに割り当てるパブリック IP アドレス (Microsoft.Network/publicIPAddresses) を作成します。ネットワーク インターフェイスは、Web サーバー用とデータベース サーバー用に 2 種類作成する必要があります。このような場合、parameters 要素とパラメーター ファイルを定義することで、1 つのテンプレート ファイルに、異なるパラメーター ファイルを使用し、Web サーバー用とデータベース サーバー用の異なるリソースを配置することができます。

1. JSON エディターに、パブリック IP アドレスおよび、ネットワーク インターフェイス リソースを作成するための以下の JSON コードを記述し、作業用フォルダーに、「**deploy-nic.json**」という名前で保存します。

```
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "subnetName": {
      "type": "string",
      "metadata": {
        "description": "Subnet Name"
      }
    },
    "dnsNameForPublicIP": {
      "type": "string",
      "metadata": {
        "description": "Unique DNS Name for the Public IP"
      }
    },
    "publicIPAddressName": {
      "type": "string",
      "metadata": {
        "description": "Public IP Address Name"
      }
    },
    "privateIPAddress": {
      "type": "string",
      "metadata": {
        "description": "Private IP Address"
      }
    },
    "nicName": {
      "type": "string",
      "metadata": {
        "description": "NIC Name"
      }
    }
  }
}
```

```

},
"variables": {
  "location": "japanwest",
  "virtualNetworkName": "AW-VNet",
  "publicIPAddressType": "Dynamic",
  "vnetID": "[resourceId('Microsoft.Network/virtualNetworks',variables('virtualNetworkName'))]",
  "subnetRef": "[concat(variables('vnetID'),'/subnets/',parameters('subnetName'))]"
},
"resources": [
  {
    "apiVersion": "2015-05-01-preview",
    "type": "Microsoft.Network/publicIPAddresses",
    "name": "[parameters('publicIPAddressName')]",
    "location": "[variables('location')]",
    "properties": {
      "publicIPAllocationMethod": "[variables('publicIPAddressType')]",
      "dnsSettings": {
        "domainNameLabel": "[parameters('dnsNameForPublicIP')]"
      }
    }
  },
  {
    "apiVersion": "2015-05-01-preview",
    "type": "Microsoft.Network/networkInterfaces",
    "name": "[parameters('nicName')]",
    "location": "[variables('location')]",
    "dependsOn": [
      "[concat('Microsoft.Network/publicIPAddresses/', parameters('publicIPAddressName'))]"
    ],
    "properties": {
      "ipConfigurations": [
        {
          "name": "ipconfig1",
          "properties": {
            "privateIPAllocationMethod": "Static",
            "privateIPAddress": "[parameters('privateIPAddress')]",
            "publicIPAddress": {
              "id": "[resourceId('Microsoft.Network/publicIPAddresses',parameters('publicIPAddressName'))]"
            },
            "subnet": {
              "id": "[variables('subnetRef')]"
            }
          }
        }
      ]
    }
  }
]
}

```

```
]
}
```

#### ワンポイント

パラメーター値は、parameter() 関数で参照します。また、concat() 関数を使用して、文字列を連結することができます。resourceId() 関数は、作成されたリソースの id を取得することができます。

2. JSON エディターに Web サーバーのネットワーク インターフェイス リソースのためのパラメーターを記述し、作業用フォルダーに、「**param-nic-for-web01.json**」という名前で保存します。

```
{
  "$schema": "http://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "subnetName": {
      "value": "Frontend-SN"
    },
    "dnsNameForPublicIP": {
      "value": "aw-web-ps"
    },
    "publicIPAddressName": {
      "value": "Web01-IP"
    },
    "privateIPAddress": {
      "value": "192.168.1.4"
    },
    "nicName": {
      "value": "Web01-NIC"
    }
  }
}
```

3. 保存したテンプレート ファイルとパラメーター ファイルを使用して、Web サーバー用の ネットワーク インターフェイスを作成するため、Windows PowerShell ISE のスクリプトを次のコードに変更 (以下の赤字部分) して、選択状態にした後、[選択項目を実行] ボタンで実行します。

```
# リソース グループ名
$rg = "AW-RG"

# テンプレート ファイルを使用した NIC とパブリック IP アドレスの作成
New-AzureRmResourceGroupDeployment -ResourceGroupName $rg `
-TemplateFile D:¥Solutions¥deploy-nic.json `
-TemplateParameterFile D:¥Solutions¥param-nic-for-web01.json
```

**ワンポイント**

このスクリプトは、JSON ファイルが D ドライブの Solutions フォルダに保存されているものとしています。異なるフォルダを使用している場合、適宜、フォルダ名を変更してください。

4. ネットワーク インターフェイスの作成が完了すると次のメッセージが返されます。

```
DeploymentName      : deploy-nic
ResourceGroupName  : AW-RG
ProvisioningState   : Succeeded
Timestamp          : 2015/12/04 2:06:58
Mode               : Incremental
TemplateLink       :
Parameters         :
                   Name      Type      Value
                   =====
subnetName         String    Frontend-SN
dnsNameForPublicIP String    aw-web-ps
publicIPAddressName String    Web01-IP
privateIPAddress   String    192.168.1.4
nicName            String    Web01-NIC

Outputs           :
```

**ワンポイント**

ProvisioningState が Succeeded になっていることを確認します。

5. 作成されたリソースを確認するため、次のコードを選択状態にして、[選択項目を実行] ボタンで実行します。

```
# 作成されたリソースの確認
Get-AzureRmResource | ? {$_.ResourceGroupName -eq $rg } | Select Name, ResourceType
```

```
Name      ResourceType
-----
Web01-NIC Microsoft.Network/networkInterfaces
Web01-IP  Microsoft.Network/publicIPAddresses
AW-VNet   Microsoft.Network/virtualNetworks
awvmstorageacct Microsoft.Storage/storageAccounts
```

6. JSON エディターで param-nic-for-web01.json をデータベース サーバーのネットワーク インターフェイスのためのパラメーターに修正 (以下の赤字部分) し、作業用フォルダに、「**param-nic-for-sql01.json**」という名前で保存します。

```
{
  "$schema": "http://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "subnetName": {
      "value": "Backend-SN"
    },
    "dnsNameForPublicIP": {
      "value": "aw-sql-ps"
    },
    "publicIPAddressName": {
      "value": "SQL01-IP"
    },
    "privateIPAddress": {
```

```

    "value": "192.168.1.36"
  },
  "nicName": {
    "value": "SQL01-NIC"
  }
}
}
}

```

7. 保存したテンプレート ファイルとパラメーター ファイルを使用して、データベース サーバー用の ネットワーク インターフェイスを作成するため、Windows PowerShell ISE で、「# 作成されたリソースの確認」のコメントの直前に、以下のコードを追加して、選択状態にした後、[選択項目を実行] ボタンで実行します。

```

New-AzureRmResourceGroupDeployment -ResourceGroupName $rg `
-TemplateFile D:\Solutions\deploy-nic.json `
-TemplateParameterFile D:\Solutions\param-nic-for-SQL01.json

```

**ワンポイント**

このスクリプトは、JSON ファイルが D ドライブの Solutions フォルダーに保存されているものとしています。異なるフォルダーを使用している場合、適宜、フォルダー名を変更してください。

8. ネットワーク インターフェイスの作成が完了すると次のメッセージが返されます。

```

DeploymentName      : deploy-nic
ResourceGroupName  : AW-RG
ProvisioningState   : Succeeded
Timestamp          : 2015/12/01 6:16:46
Mode               : Incremental
TemplateLink       :
Parameters         :
                    Name                Type                Value
                    -----
                    subnetName          String              Backend-SN
                    dnsNameForPublicIP  String              aw-sql-ps
                    publicIPAddressName String              SQL01-IP
                    privateIPAddress    String              192.168.1.36
                    nicName             String              SQL01-NIC

Outputs            :

```

**ワンポイント**

ProvisioningState が Succeeded になっていることを確認します。



9. 作成されたリソースを確認するため、次のコードを選択状態にして、[選択項目を実行] ボタンで実行します。

```
# 作成されたリソースの確認
```

```
Get-AzureRmResource | ? {$_.ResourceGroupName -eq $rg } | Select Name, ResourceType
```

```
Name                ResourceType
----                -
SQL01-NIC           Microsoft.Network/networkInterfaces
Web01-NIC           Microsoft.Network/networkInterfaces
SQL01-IP            Microsoft.Network/publicIPAddresses
Web01-IP            Microsoft.Network/publicIPAddresses
AW-VNet             Microsoft.Network/virtualNetworks
awvmstorageacct    Microsoft.Storage/storageAccounts
```

10. Windows PowerShell ISE の [ファイル] メニューから [名前を付けて保存] を選択し、作成したスクリプトを作業用フォルダーに、「**Create-NIC-ByTemplate.ps1**」という名前で保存します。作成したスクリプトは、以下のコードになります。

```
# リソース グループ名
```

```
$rg = "AW-RG"
```

```
# テンプレート ファイルを使用した NIC とパブリック IP アドレスの作成
```

```
New-AzureRmResourceGroupDeployment -ResourceGroupName $rg `
```

```
-TemplateFile D:¥Solutions¥deploy-nic.json `
```

```
-TemplateParameterFile D:¥Solutions¥param-nic-for-web01.json
```

```
New-AzureRmResourceGroupDeployment -ResourceGroupName $rg `
```

```
-TemplateFile D:¥Solutions¥deploy-nic.json `
```

```
-TemplateParameterFile D:¥Solutions¥param-nic-for-SQL01.json
```

```
# 作成されたリソースの確認
```

```
Get-AzureRmResource | ? {$_.ResourceGroupName -eq $rg } | Select Name, ResourceType
```

## 4.5 Azure 仮想マシンの作成

ここでは、テンプレート ファイルを使用して仮想マシン (Microsoft.Compute/virtualMachines) を作成します。仮想マシンのプロパティ (properties) には、下表のプロファイルを定義する必要があります。

プロファイル名	説明
<b>hardwareProfile</b>	仮想マシン サイズを指定します。
<b>osProfile</b>	コンピューター名、管理者アカウント、管理者パスワードを指定します。
<b>storageProfile</b>	imageReference に、ソースとして使用するイメージの情報 (publisher、offer、sku、version) を指定し、osDisk に、作成した仮想マシンの OS ディスクを保存する Blob ストレージのパスを指定します。
<b>networkProfile</b>	仮想マシンで、使用する ネットワーク インターフェイスの ID を resourceId 関数で取得します。
<b>diagnosticsProfile</b>	起動の診断の有効化と使用するストレージを指定します。

1. JSON エディターに、パブリック IP アドレスおよび、ネットワーク インターフェイス リソースを作成するための以下の JSON コードを記述し、作業用フォルダーに、「**deploy-vm.json**」という名前で保存します。

```
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "nicName": {
      "type": "string",
      "metadata": {
        "description": "NIC Name"
      }
    },
    "vmName": {
      "type": "string",
      "metadata": {
        "description": "VM Name"
      }
    },
    "vmSize": {
      "type": "string",
      "metadata": {
        "description": "VM Size"
      }
    },
    "publisher": {
      "type": "string",
      "metadata": {
        "description": "Publisher for the VM Image"
      }
    },
    "offer": {
      "type": "string",
```

```

"metadata": {
  "description": "Offer for the VM Image"
},
"sku": {
  "type": "string",
  "metadata": {
    "description": "SKU for the VM Image"
  }
},
"OSDiskName": {
  "type": "string",
  "metadata": {
    "description": "OS Disk Name"
  }
},
"variables": {
  "location": "japanwest",
  "storageAccountName": "awvmstorageacct",
  "storageContainerName": "vhds",
  "adminUsername": "awadmin",
  "adminPassword" : "awPa$$w0rd",
  "storageUri": "[concat('http://',variables('storageAccountName'),'blob.core.windows.net')]",
  "vhdUri": "[concat('/',variables('storageContainerName'),'/',parameters('OSDiskName'),'vhd')]"
},
"resources": [
{
  "apiVersion": "2015-06-15",
  "type": "Microsoft.Compute/virtualMachines",
  "name": "[parameters('vmName')]",
  "location": "[variables('location')]",
  "properties": {
    "hardwareProfile": {
      "vmSize": "[parameters('vmSize')]"
    },
    "osProfile": {
      "computerName": "[parameters('vmName')]",
      "adminUsername": "[variables('adminUsername')]",
      "adminPassword": "[variables('adminPassword')]"
    },
    "storageProfile": {
      "imageReference": {
        "publisher": "[parameters('publisher')]",
        "offer": "[parameters('offer')]",
        "sku": "[parameters('sku')]",
        "version": "latest"
      }
    }
  }
}
]

```

```
    },
    "osDisk": {
      "name": "osdisk",
      "vhd": {
        "uri": "[concat(variables('storageUri') ,variables('vhdUri'))]"
      },
      "caching": "ReadWrite",
      "createOption": "FromImage"
    }
  },
  "networkProfile": {
    "networkInterfaces": [
      {
        "id": "[resourceId('Microsoft.Network/networkInterfaces',parameters('nicName'))]"
      }
    ]
  },
  "diagnosticsProfile": {
    "bootDiagnostics": {
      "enabled": "true",
      "storageUri": "[variables('storageUri')]"
    }
  }
}
]
```

2. JSON エディターで以下の Web サーバーの仮想マシンのためのパラメーターを入力し、作業用フォルダーに「**param-vm-for-web01.json**」という名前で保存します。

```
{
  "$schema": "http://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "nicName": {
      "value": "Web01-NIC"
    },
    "vmName": {
      "value": "Web01"
    },
    "vmSize": {
      "value": "Standard_A1"
    },
    "publisher": {
      "value": "MicrosoftWindowsServer"
    },
    "offer": {
      "value": "WindowsServer"
    },
    "sku": {
      "value": "2012-R2-Datacenter"
    },
    "OSDiskName": {
      "value": "Web01"
    }
  }
}
```

3. 保存したテンプレート ファイルとパラメーター ファイルを使用して、Web サーバー用の仮想マシンを作成するため、Windows PowerShell ISE のスクリプトを次のコードに変更して、選択状態にした後、[選択項目を実行] ボタンで実行します。

```
# リソース グループ名

$rg = "AW-RG"

# テンプレート ファイルを使用した仮想マシンの作成

New-AzureRmResourceGroupDeployment -ResourceGroupName $rg `
-TemplateFile D:\Solutions\deploy-Vm.json `
-TemplateParameterFile D:\Solutions\param-vm-for-web01.json
```

#### ワンポイント

このスクリプトは、JSON ファイルが D ドライブの Solutions フォルダに保存されているものとしています。異なるフォルダを使用している場合、適宜、フォルダ名を変更してください。

4. 仮想マシンの作成が完了すると次のメッセージが返されます。

```
DeploymentName      : deploy-vm
ResourceGroupName  : AW-RG
ProvisioningState   : Succeeded
Timestamp          : 2015/12/01 6:41:21
Mode               : Incremental
TemplateLink       :
Parameters         :
                    Name                Type                Value
                    -----
                    nicName             String              Web01-NIC
                    vmName              String              Web01
                    vmSize              String              Standard_A1
                    publisher           String              MicrosoftWindowsServer
                    offer                String              WindowsServer
                    sku                  String              2012-R2-Datacenter
                    osDiskName          String              Web01

Outputs            :
```

#### ワンポイント

ProvisioningState が Succeeded になっていることを確認します。

5. 作成されたリソースを確認するため、次のコードを選択状態にして、[選択項目を実行] ボタンで実行します。

```
# 作成されたリソースの確認

Get-AzureRmResource | ? {$_.ResourceGroupName -eq $rg } | Select Name, ResourceType
```

```
Name                ResourceType
----                -
Web01               Microsoft.Compute/virtualMachines
SQL01-NIC           Microsoft.Network/networkInterfaces
Web01-NIC           Microsoft.Network/networkInterfaces
SQL01-IP            Microsoft.Network/publicIPAddresses
Web01-IP            Microsoft.Network/publicIPAddresses
AW-VNet             Microsoft.Network/virtualNetworks
awvmstorageacct    Microsoft.Storage/storageAccounts
```

11. JSON エディターで param-vm-for-web01.json をデータベース サーバーの仮想マシンのためのパラメーターに修正 (以下の赤字部分) し、作業用フォルダーに、「**param-vm-for-sql01.json**」という名前で保存します。

```
{
  "$schema": "http://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "nicName": {
      "value": "SQL01-NIC"
    },
    "vmName": {
      "value": "SQL01"
    },
    "vmSize": {
      "value": "Standard_A1"
    },
    "publisher": {
      "value": "MicrosoftSQLServer"
    },
    "offer": {
      "value": "SQL2014SP1-WS2012R2"
    },
    "sku": {
      "value": "Enterprise"
    },
    "OSDiskName": {
      "value": "SQL01"
    }
  }
}
```

12. 保存したテンプレート ファイルとパラメーター ファイルを使用して、データベース サーバー用の仮想マシンを作成するため、Windows PowerShell ISE で、「# 作成されたリソースの確認」のコメントの直前に、以下のコードを追加して、選択状態にした後、[選択項目を実行] ボタンで実行します。

```
New-AzureRmResourceGroupDeployment -ResourceGroupName $rg `
-TemplateFile D:¥Solutions¥deploy-vm.json `
-TemplateParameterFile D:¥Solutions¥param-vm-for-sql01.json
```

#### ワンポイント

このスクリプトは、JSON ファイルが D ドライブの Solutions フォルダーに保存されているものとしています。異なるフォルダーを使用している場合、適宜、フォルダー名を変更してください。

13. 仮想マシンの作成が完了すると次のメッセージが返されます。

```
DeploymentName      : deploy-vm
ResourceGroupName  : AW-RG
ProvisioningState  : Succeeded
Timestamp          : 2015/12/01 7:15:15
Mode               : Incremental
TemplateLink       :
Parameters         :
                   Name      Type      Value
                   =====
                   nicName    String    SQL01-NIC
                   vmName     String    SQL01
                   vmSize     String    Standard_A1
                   publisher  String    MicrosoftSQLServer
                   offer       String    SQL2014SP1-WS2012R2
                   sku         String    Enterprise
                   osDiskName String    SQL01

Outputs           :
```

**ワンポイント**  
ProvisioningState が Succeeded になっていることを確認します。

14. 作成されたリソースを確認するため、次のコードを選択状態にして、[選択項目を実行] ボタンで実行します。

```
# 作成されたリソースの確認
Get-AzureRmResource | ? {$_.ResourceGroupName -eq $rg } | Select Name, ResourceType
```

```
Name      ResourceType
-----
SQL01     Microsoft.Compute/virtualMachines
Web01     Microsoft.Compute/virtualMachines
SQL01-NIC Microsoft.Network/networkInterfaces
Web01-NIC Microsoft.Network/networkInterfaces
SQL01-IP  Microsoft.Network/publicIPAddresses
Web01-IP  Microsoft.Network/publicIPAddresses
AW-VNet   Microsoft.Network/virtualNetworks
awvmstorageacct Microsoft.Storage/storageAccounts
```

15. Windows PowerShell ISE の [ファイル] メニューから [名前を付けて保存] を選択し、作成したスクリプトを作業用フォルダーに、「**Create-vm-ByTemplate.ps1**」という名前で保存します。作成したスクリプトは、以下のコードになります。

```
# リソース グループ名
$rg = "AW-RG"

# テンプレート ファイルを使用した仮想マシンの作成
New-AzureRmResourceGroupDeployment -ResourceGroupName $rg `
-TemplateFile D:\%Solutions%\deploy-vm.json `
-TemplateParameterFile D:\%Solutions%\param-vm-for-web01.json

New-AzureRmResourceGroupDeployment -ResourceGroupName $rg `
-TemplateFile D:\%Solutions%\deploy-vm.json `
-TemplateParameterFile D:\%Solutions%\param-vm-for-sql01.json
```



```
# 作成されたリソースの確認
```

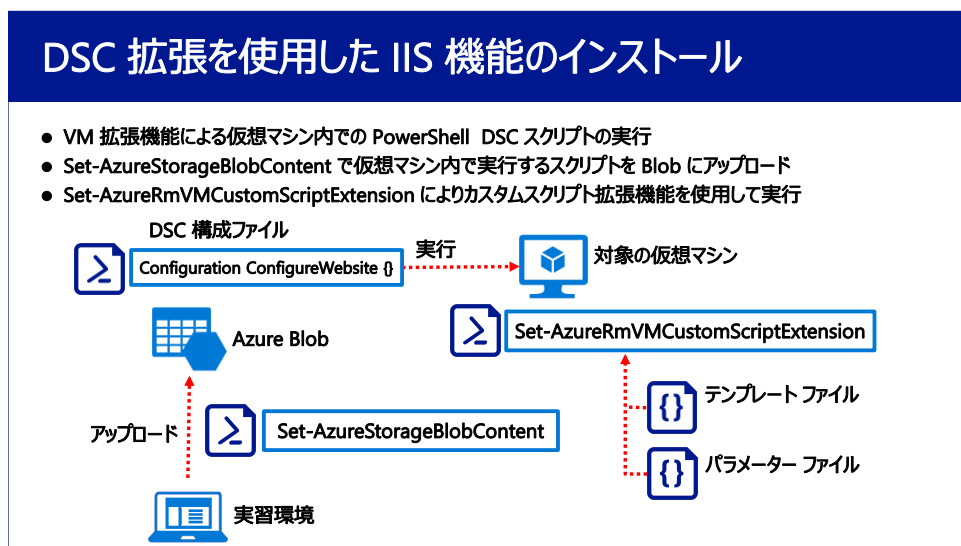
```
Get-AzureRmResource | ? { $_.ResourceGroupName -eq $rg } | Select Name, ResourceType
```

## 4.6 DSC 拡張機能を使用した IIS 機能のインストール

DSC (Desired State Configuration) は Windows PowerShell の新しい管理プラットフォームで、宣言型 PowerShell 構文を使用して、物理サーバー、および仮想マシンの構成管理を行うことができます。DSC を使用すると、リモートから対象ノードに対して、以下のような操作が可能です。

- サーバー ロールと機能の有効化と無効化
- レジストリ設定の管理
- ファイルとディレクトリの管理
- プロセスとサービスの開始、停止、および管理
- グループとユーザー アカウントの管理
- 新しいソフトウェアの配置
- 環境変数の管理
- Windows PowerShell スクリプトの実行
- 必要な状態とは異なる構成の修正
- 特定のノードでの実際の構成状態の検出

ここでは、Azure 仮想マシンの DSC 拡張機能を使用し、前の手順で作成した Web サーバーに IIS をインストールします。DSC 構成ファイルは、ローカル コンピューターで PowerShell スクリプトとして作成し、Azure Blob ストレージに圧縮ファイル (ZIP) 形式にしてアップロードします。JSON テンプレートに DSC 構成ファイルを仮想マシンに適用する「Microsoft.Compute/virtualMachines/extensions」リソースを定義して、New-AzureRmResourceGroupDeployment コマンドレットで実行することができます。



1. 最初に IIS 関連の機能をインストールする DSC 構成ファイルを作成するため [ファイル] メニューから [新規作成] を選択して、新しいタブを開き、以下のコードを入力します。

```
Configuration ConfigureWebsite
{
    param ($MachineName)
    Node $MachineName
    {
        ### IIS
        WindowsFeature IIS
        {
            Name = "Web-Server"
            Ensure = "Present"
        }
        ### ASP.NET 4.5
        WindowsFeature AspNet45
        {
            Name = "Web-Asp-Net45"
            Ensure = "Present"
        }
        ### Management Service
        WindowsFeature ManagementService
        {
            Name = "Web-Mgmt-Service"
            Ensure = "Present"
        }
    }
}
```

#### ワンポイント

この DSC 構成ファイルには、3 つの構成ブロックが定義されています。このファイルを対象ノードの Windows Server 仮想マシン インスタンスに適用することで、IIS、ASP.NET 4.5、管理サービスの各機能をインストールすることができます。

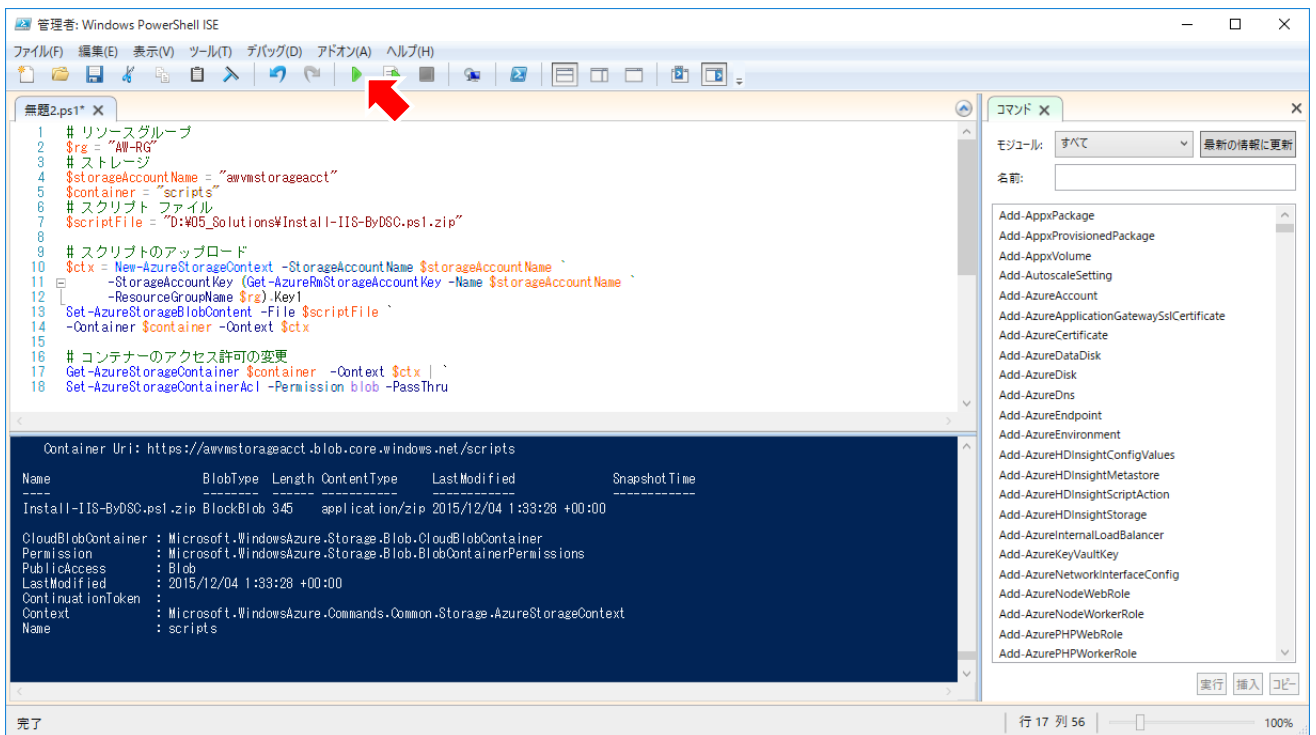
2. [ファイル] メニューから [名前を付けて保存] を選択し、作成したスクリプトを作業用フォルダーに、「**Install-IIS-ByDSC.ps1**」という名前で保存します。
3. [Install-IIS-ByDSC.ps1] タブを閉じます。
4. Windows エクスプローラーから Install-IIS-ByDSC.ps1 を右クリックして [送る] → [圧縮 (zip 形式) フォルダー] を選択して ZIP ファイルにした後、ファイル名を「**Install-IIS-ByDSC.ps1.zip**」に変更します。

5. 作成したスクリプトを Blob ストレージに保存するため、Windows PowerShell ISE の [ファイル] メニューから [新規作成] を選択して、新しいタブを開き、以下のコードを入力して、[スクリプトを実行] ボタンをクリックして、実行します。

```
# リソースグループ
$rg = "AW-RG"
# ストレージ
$storageAccountName = "awvmstorageacct"
$container = "scripts"
# スクリプト ファイル
$scriptFile = "D:\%Solutions%\Install-IIS-ByDSC.ps1.zip"

# スクリプトのアップロード
$ctx = New-AzureStorageContext -StorageAccountName $storageAccountName `
    -StorageAccountKey (Get-AzureRmStorageAccountKey -Name $storageAccountName `
    -ResourceGroupName $rg).Key1
Set-AzureStorageBlobContent -File $scriptFile `
    -Container $container -Context $ctx

# コンテナのアクセス許可の変更
Get-AzureStorageContainer $container -Context $ctx | `
Set-AzureStorageContainerAcl -Permission blob -PassThru
```



**ワンポイント**

このスクリプトにより、指定したストレージ アカウントの scripts コンテナに、前の手順で作成した「Install-IIS-ByDSC.ps1.zip」が、アップロードされます。本書では、DSC 構成ファイルを対象の仮想マシンにダウンロードする際に、匿名アクセスを使用しますので、scripts コンテナのアクセス許可を「public blob」に変更しています。

なお、このスクリプトでは、「Install-IIS-ByDSC.ps1.zip」は、D ドライブの Solutions フォルダに置かれているものとしています。ファイル パスは、ご自身の実習環境に合うように修正してください。

6. [ファイル] メニューから [名前を付けて保存] を選択し、作成したスクリプトを作業用フォルダに、「**Upload-DSCScript.ps1**」という名前で保存します。
7. [Upload-DSCScript.ps1] タブを閉じます。
8. Blob ストレージにアップロードした DSC 構成を Web01 に適用するための以下の JSON コードを JSON エディターで記述し、作業用フォルダに「**install-iis.json**」という名前で保存します。

```
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "vmName": {
      "type": "string",
      "metadata": {
        "description": "Name of the VM"
      }
    },
  },
  "configurationFunction": {
    "type": "string",
    "defaultValue": "Install-IIS-ByDSC.ps1¥¥ConfigureWebsite",
    "metadata": {
      "description": "DSC configuration function to call"
    }
  },
  "variables": {
    "location": "japanwest",
    "vmExtensionName": "dscExtension",
    "storageAccountName": "awvmstorageacct",
    "storageContainerName": "scripts",
    "storageUri": "[concat('https://',variables('storageAccountName'),'blob.core.windows.net')]",
    "scriptUri": "[concat('/',variables('storageContainerName'),'Install-IIS-ByDSC.ps1.zip')]"
  },
  "resources": [
    {
      "type": "Microsoft.Compute/virtualMachines/extensions",
      "name": "[concat(parameters('vmName'),'/', variables('vmExtensionName'))]",
      "apiVersion": "2015-05-01-preview",
      "location": "[variables('location')]",
      "properties": {
        "publisher": "Microsoft.Powershell",
```

```

    "type": "DSC",
    "typeHandlerVersion": "2.8",
    "settings": {
      "ModulesUri": "[concat(variables('storageUri') ,variables('scriptUri'))]",
      "ConfigurationFunction": "[parameters('configurationFunction')]",
      "Properties": {
        "MachineName": "[parameters('vmName')]"
      }
    },
    "protectedSettings": null
  }
}
]
}

```

16. JSON エディターに Web サーバーの ネットワーク インターフェイスのためのパラメーターを記述し、作業用フォルダーに、「**param-nic-for-web01.json**」という名前で保存します。

```

"$schema": "http://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",
"contentVersion": "1.0.0.0",
"parameters": {
  "vmName": {
    "value": "Web01"
  },
  "configurationFunction": {
    "value": "Install-IIS-ByDSC.ps1¥¥ConfigureWebsite"
  }
}
}
}

```

17. 保存したテンプレート ファイルとパラメーター ファイルを使用して、DSC 構成を仮想マシン インスタンスにダウンロードして適用するため、Windows PowerShell ISE に以下のスクリプトを入力して [スクリプトを実行] ボタンで実行します。

```

# リソース グループ名
$rg = "AW-RG"

# DSC 拡張機能を使用した IIS のインストール
New-AzureRmResourceGroupDeployment -ResourceGroupName $rg `
-TemplateFile D:¥Solutions¥install-iis.json `
-TemplateParameterFile D:¥Solutions¥param-iis-for-web01.json

```

#### ワンポイント

このスクリプトは、JSON ファイルが D ドライブの Solutions フォルダーに保存されているものとしています。異なるフォルダーを使用している場合、適宜、フォルダー名を変更してください。

18. IIS のインストールが完了すると次のメッセージが返されます。

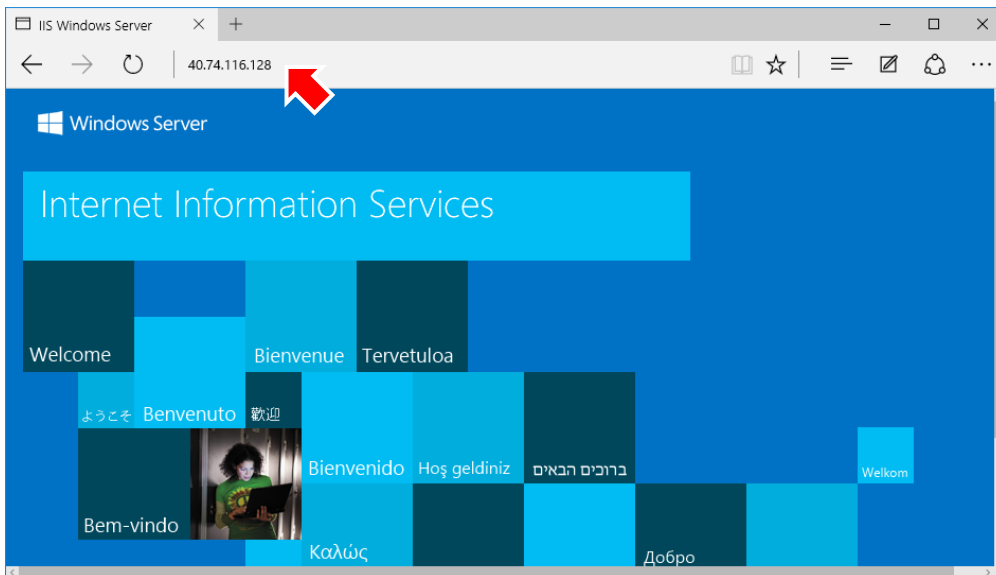
```
DeploymentName : install-iis
ResourceGroupName : AW-RG
ProvisioningState : Succeeded
Timestamp : 2015/12/04 0:12:02
Mode : Incremental
TemplateLink :
Parameters :
    Name      Type      Value
    =====  =====  =====
    vmName    String   Web01
    configurationFunction String   Install-IIS-ByDSC.ps1#ConfigureWebsite

Outputs :
```

#### ワンポイント

このスクリプトの実行には、10 分程度の時間がかかります。このスクリプトにより、DSC 構成に記述された内容が、仮想マシンに適用され、IIS 機能がインストールされます。正常に終了すると ProvisioningState が Succeeded で返されます。

9. Web ブラウザーを起動して、アドレス バーに、「<http://<Web01 のパブリック IP アドレス>>」と入力して IIS の既定のページが参照できることを確認します。



#### ワンポイント

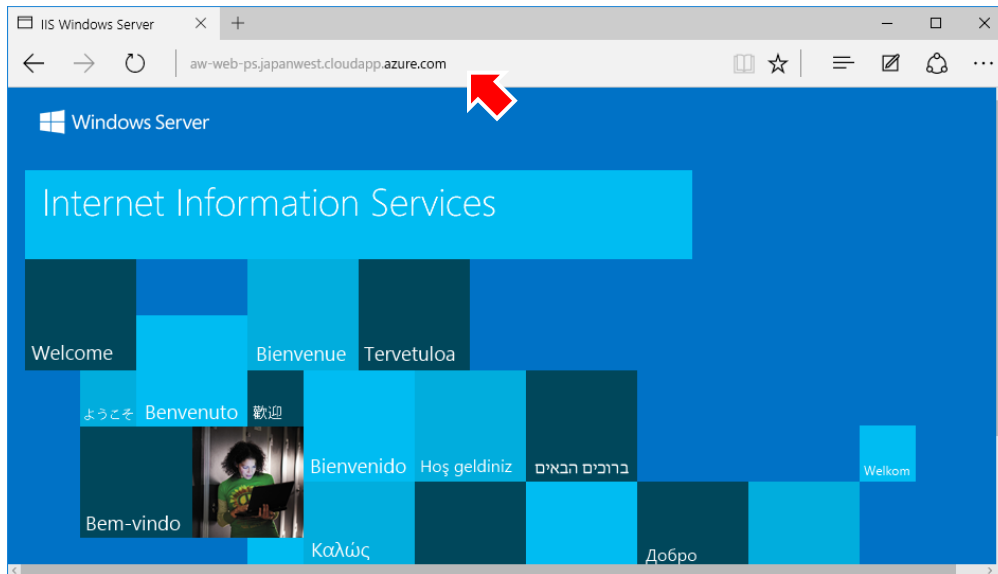
パブリック IP アドレスの確認は、Get-AzureRmPublicIpAddress コマンドレットを使用します。

```
Get-AzureRmPublicIpAddress -ResourceGroupName $rg | `
select Name,IpAddress
```

```
PS C:\Windows\system32> Get-AzureRmPublicIpAddress -ResourceGroupName $rg | `
select Name,IpAddress

Name      IPAddress
-----
SQL01-IP  40.74.118.133
Web01-IP  40.74.118.128
```

- 次に、パブリック IP アドレス リソースに割り当てた ドメイン名ラベルを使用して IIS にアクセスするために Web ブラウザーのアドレス バーに、「<http://aw-web-ps.japanwest.cloudapp.azure.com>」と入力して既定のページが参照できることを確認します。



#### ワンポイント

ドメイン名ラベルの「aw-web-ps」は、「4.4 ネットワーク インターフェイスとパブリック IP アドレスの作成」で、ご自身が `dnsNameForPublicIP` パラメーターに、ご自身が設定したラベルに変更して実行してください。

- [ファイル] メニューから [名前を付けて保存] を選択し、作成したスクリプトを作業用フォルダーに、「**Run-DSCScript.ps1**」という名前で保存します。
- [Run-DSCScript.ps1] タブを閉じます。



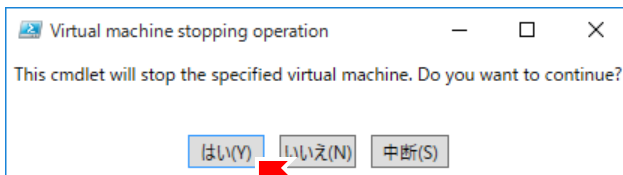
## 4.7 仮想マシンの停止

仮想マシン インスタンスはプロビジョニングが完了すると インスタンスが稼働状態となり、課金が発生します。仮想マシン インスタンスは、Stop-AzureRmVM を使用して、リソースを解放した状態で停止することができます。ここでは、指定したリソース グループに含まれるすべての仮想マシン インスタンスを停止する方法を説明します。

1. Windows PowerShell ISE の [表示] メニューの [スクリプト ウィンドウの表示] を選択して、新しいスクリプト ウィンドウを開きます。
2. スクリプト ウィンドウに以下のコードを記述した後、[スクリプトを実行] ボタンをクリックして、実行します。

```
# リソースグループ
$rg = "AW-RG"
# 仮想マシンの停止
Get-AzureRmVM -ResourceGroupName $rg | select name | Stop-AzureRmVM -ResourceGroupName $rg
```

3. 以下のメッセージが表示されたら、[はい] ボタンをクリックして、仮想マシン インスタンスを停止します。



```
PS C:\Windows\system32> # リソースグループ
$rg = "AW-RG"
# 仮想マシンの停止
Get-AzureRmVM -ResourceGroupName $rg | select name | Stop-AzureRmVM -ResourceGroupName $rg

Status           : Succeeded
StatusCode        : OK
RequestId         : 6a290fa3-eddd-4069-a040-9138f90b3b88
Output            :
Error             :
StartTime         : 2015/12/04 12:00:54 +09:00
EndTime           : 2015/12/04 12:00:58 +09:00
TrackingOperationId : c28be9f0-0c67-464a-9e3c-9d42935d071b

Status           : Succeeded
StatusCode        : OK
RequestId         : fd7e55ea-a9a7-4b37-803a-55d3c3a17af0
Output            :
Error             :
StartTime         : 2015/12/04 12:01:45 +09:00
EndTime           : 2015/12/04 12:01:48 +09:00
TrackingOperationId : 1f5ab5f6-41a6-4f43-8b35-28be9b28e85c
```

### ワンポイント

Status が Succeeded になることを確認します。このスクリプトにより、指定したリソース グループに含まれるすべての仮想マシン が列挙され、パイプ処理により、1 台ずつインスタンスがシャットダウンされます。なお、停止している仮想マシン インスタンスの起動は、Start-AzureRmVM コマンドレットを使用します。また、仮想マシン インスタンスを再起動する場合は、Restart-AzureRmVM コマンドレットを使用することができます。

4. [ファイル] メニューから [名前を付けて保存] を選択し、作成したスクリプトを作業用フォルダーに、「**Stop-VM.ps1**」という名前で保存します。
5. [Stop-VM.ps1] タブを閉じます。

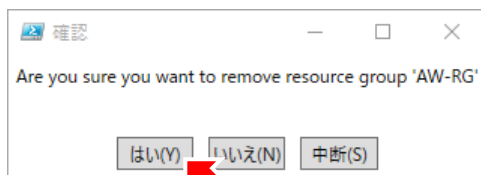
## 4.8 作成したリソース グループの削除

ここでは、作成したリソース グループを削除します。個々のリソースを個別に削除することもできますが、リソース グループを削除することで、そのリソース グループ内のすべてのリソースを削除することができます。AW-RG リソース グループを削除することで、その中にあるリソースをすべてが削除されます。また、検証を続けたい場合は、検証の完了後に行ってください。

1. Windows PowerShell ISE の [ファイル] メニューから [開く] を選択し、3 章で作成した「Remove-ResourceGroup.ps1」を開きます。スクリプトは、以下のコードになります。

```
リソースグループ  
$rg = "AW-RG"  
  
# リソース グループの削除  
Remove-AzureRmResourceGroup -Name $rg  
# リソースグループの確認  
Get-AzureRmResourceGroup | Select ResourceGroupName
```

2. [スクリプトを実行] ボタンをクリックして実行し、リソース グループを削除します。以下の確認メッセージが表示されたら [はい] ボタンをクリックします。



### ワンポイント

表示されるリストに、「AW-RG」が存在しないことを確認します。

3. [Remove-ResourceGroup.ps1] タブを閉じます。

## 4. Microsoft Azure に関する情報の入手元

Microsoft Azure に関する最新の情報は、次の Web サイトから入手できます。



- Azure の公式ページ (各国共通)

製品情報、価格、技術情報など、Azure に関するすべての情報への入口です。Azure をお使いのお客様は、右上の「ポータル」をクリックすると Azure のポータルにログインすることができます。ポータルでは、Azure 上で構築したアプリケーションの管理、課金状況の確認などができます。

<http://azure.microsoft.com/ja-jp/>

- 日本のお客様向けのサイト

上記サイトの内容に加え、日本のお客様用に作成されたコンテンツが満載です。

<http://www.microsoft.com/ja-jp/server-cloud/azure/Explore.aspx>

- サポートエンジニアによるブログ サイト

よくお問い合わせをいただく技術・課金・サポートに関する内容をまとめたものです。サポートに問い合わせる前に、まずはここをご参照ください。

<http://blogs.msdn.com/b/dsazurejp/>

<http://blogs.msdn.com/b/jpsql/> (Azure SQL データベース / Server Server)

- MSDN フォーラムの Azure フォーラム

Azure に関する技術的な質問に対して、これまでの投稿から情報を入手することができます。また、自分で質問を投稿し、他のユーザーや MVP (Most Valuable Professional) からの回答を得ることが期待できます。開発者 (DEVELOPER) サポート / 標準 (STANDARD) サポート契約ではカバーしていない “How To” や “仕様” に関する質問も、このフォーラムをご活用ください。

<https://social.msdn.microsoft.com/Forums/ja-JP/home?category=azure>

## 5. Microsoft Azure のお問合せ

Microsoft Azure では、お問い合わせ内容に応じて窓口を用意しております。お問い合わせの内容に応じて各窓口をご利用ください。なお、Premier のお客様は、Premier 窓口をご利用いただけます。

お問合せ内容	料金	窓口名	連絡方法	お問い合わせ方法
製品や機能の概要、価格、ライセンスなどの情報の収集や購入前相談	無償	Cloud Direct	電話、メール	「Cloud Direct」で検索してください。
課金、サブスクリプションに関するお問い合わせ、請求書払いへの変更、クォータ増加の依頼	無償	Microsoft Azure 新ポータル	Azure 新ポータル ※1	※3
技術的なお問い合わせ	有償	Microsoft Azure 新ポータル ※2		

※1 : 電話窓口はありません。

※2 : 有償のサポートプランが必要です。

※3 : お問い合わせ方法につきましては、次の Blog をご参照ください。

サポートにお問い合わせする方法について

<http://blogs.msdn.com/b/dsazurejp/archive/2013/10/31/10462044.aspx>

なお、ポータルサイトにアクセスできない場合は、次のどちらかの方法でお問い合わせください。

窓口名 / 連絡方法	お問い合わせ方法
カスタマー インフォメーションセンター 電話番号 : 0120-41-6755	営業時間 : 平日 9:00 - 18:00 窓口担当者に、「Azure について問い合わせがしたい」とお伝えください。Azure 担当者より折り返しご連絡いたします。
Web フォーム <a href="http://azure.microsoft.com/ja-jp/support/options/">http://azure.microsoft.com/ja-jp/support/options/</a>	「アカウントのサインインまたはサインアップで問題が発生する場合」からお問い合わせを発行してください。