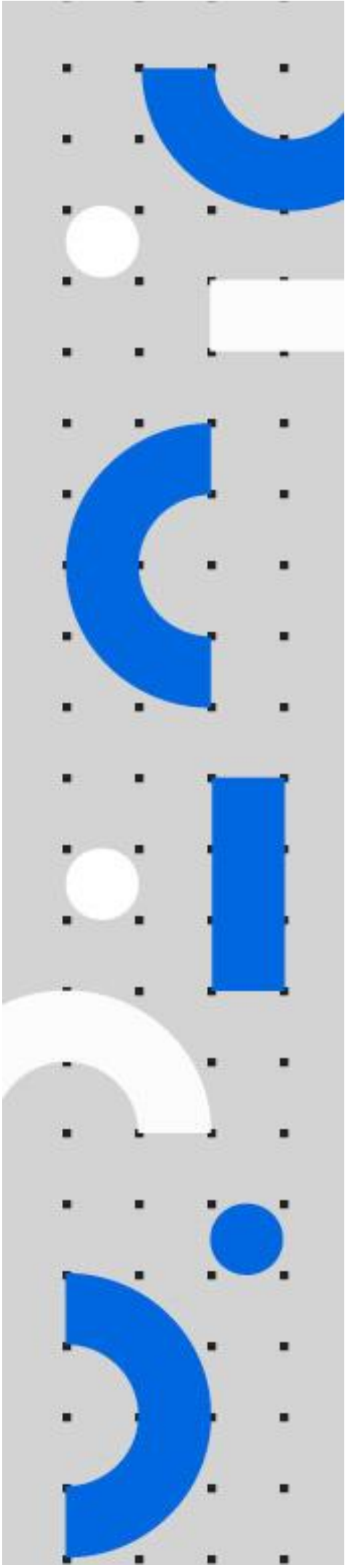
Stack-UiPath on Azure

v2018.4

v2019.10



リビジョン履歴

Date	Version	Author	Description
20th June 2020	2018.4 2019.10	Yosuke Kajimoto	First version for v2018.4 / v2019.10 release
28th June 2020	2018.4 2019.10	Yotaro Ebina	Added Elasticsearch/Kibana v6.8 / v7.6 installation with Application Gateway (Standard v2 tier) Settings
5th July 2020	2018.4 2019.10	Yosuke Kajimoto	Add contents

商標について

- UiPath のソフトウェア、製品、サービス、(これには、UiPath Orchestrator、 UiPath Robot、 UiPath Studio が含まれますが、これらに限りません) はアメリカ合衆国で登録された UiPath Inc.、 および 他の国・地域で登録された UiPath の関係会社の商標または登録商標です。 UiPath のロゴは UiPath Inc., が所有するものであり、UiPath の事前の明示的な許可なく、お客様及びその他の方が使用することはできません。
- Microsoft のソフトウェア、製品、サービス (これには、 Microsoft、 Windows、 Windows Server、 SQL Server 及び Active Directory が含まれますが、これらに限りません) は アメリカ合衆国で登録された Microsoft Corporation 及び他の国・地域で登録されたその関係会社の商標または登録商標です。
- Oracle のソフトウェア、製品、サービス (これには、 Java も含まれますがこれに限りません) は アメリカ合衆国で登録された Oracle 及びその他の国・地域で登録された関係会社の商標または登録商標です。
- Elastic は、 Elastic N.V. 及びその関係会社の商標または登録商標です。
- Redis は、 Redis Labs Ltd の商標です。
- その他、記載されている製品名、会社名およびサービス名はそれぞれの各社の商標または登録商標です。

免責事項

- 本ガイドの内容は 2020 年 7 月現在の情報であり、下記の製品リリースに基づいております。
  - UiPath Orchestrator v2018.4
  - UiPath Orchestrator v2019.10

- 製品の新しいリリース、修正プログラムなどによって、本ガイドの説明と異なる動作・仕様となる可能性がありますので、予めご留意ください。
- 本ガイドに含まれる情報は可能な限り正確を期しておりますが、 UiPath 株式会社の正式なドキュメントではありません。本ガイドに記載された内容に関して UiPath 株式会社は何ら保証していません。従って、本ガイドに含まれる情報の利用はお客様の責任においてなされるものであり、UiPath はガイドの内容によって受けたいかなる被害に関して一切の補償をするものではございません。
- 本ガイドは UiPath を法的に拘束する書類ではありません。UiPath はお客様に通知なくして、本ガイドの内容の一部または全部を修正及びアップデートできます。
- お客様は UiPath および執筆者の書面の承諾なしで本ガイドを複製、修正、頒布できません。

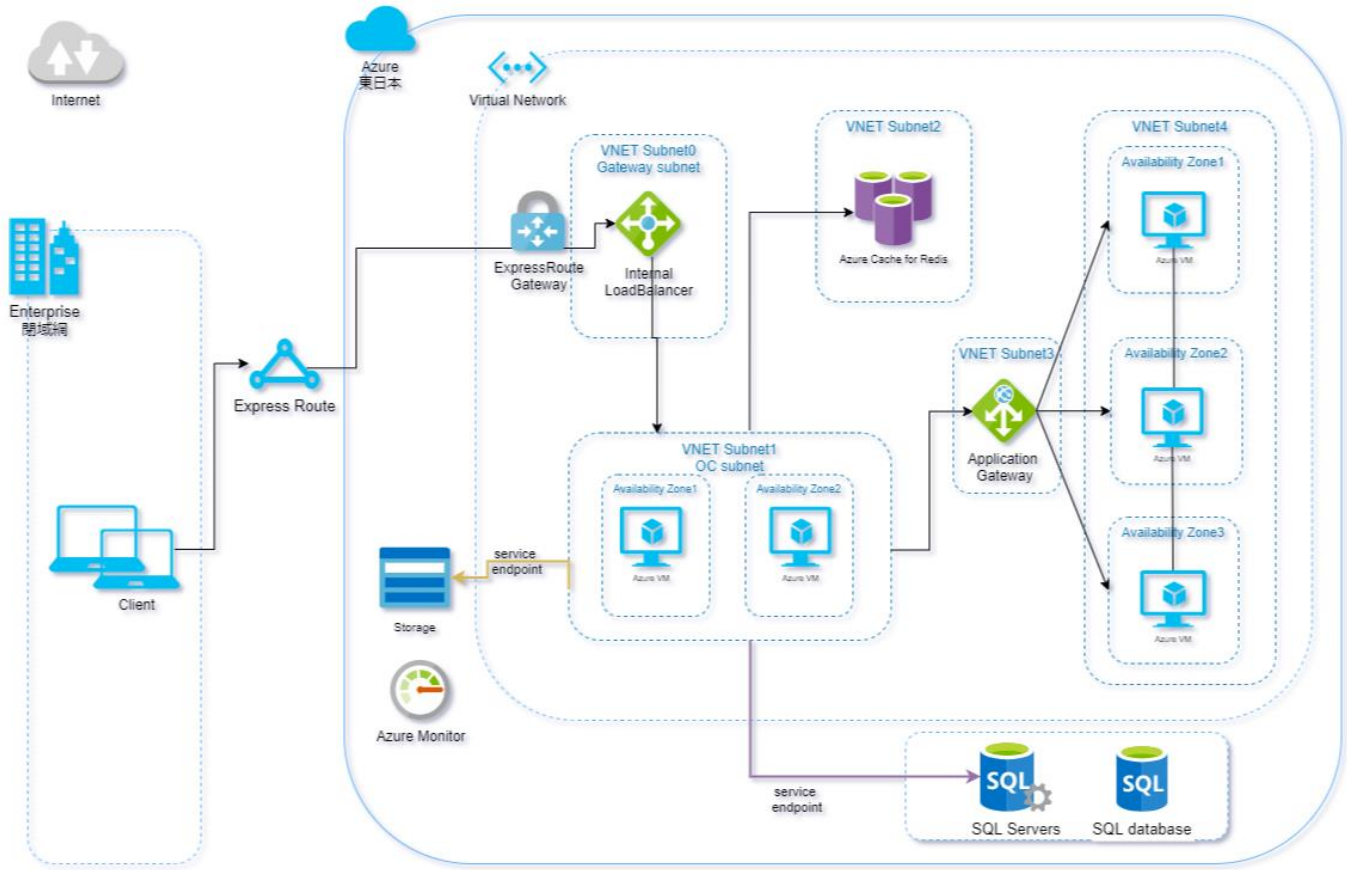
目次

0. 前提

事前に Microsoft Azure のアカウント作成及びサブスクリプションの設定が有効化されている事を前提とします。
本手順は UiPath K.K から提供される ARM テンプレート（StackUiPath.json）を利用して各 Azure リソースをプロビジョニングし、閉域網に対応した構成手順となります。
環境構築作業者は UiPath Orchestrator の知見に加え、Microsoft Azure の知見を有している事が望ましいです。
また、事前に Azure リソースグループをご準備ください。
本手順における動作確認及び、各画面キャプチャは仮想ネットワーク内に Bastion を構築し、Bastion 経由で動作確認を実施したものとなります。

1. 構成

本ソリューションの構成は下記を想定



2. Azure 利用サービス一覧

本ソリューションにおける Azure 利用サービス一覧

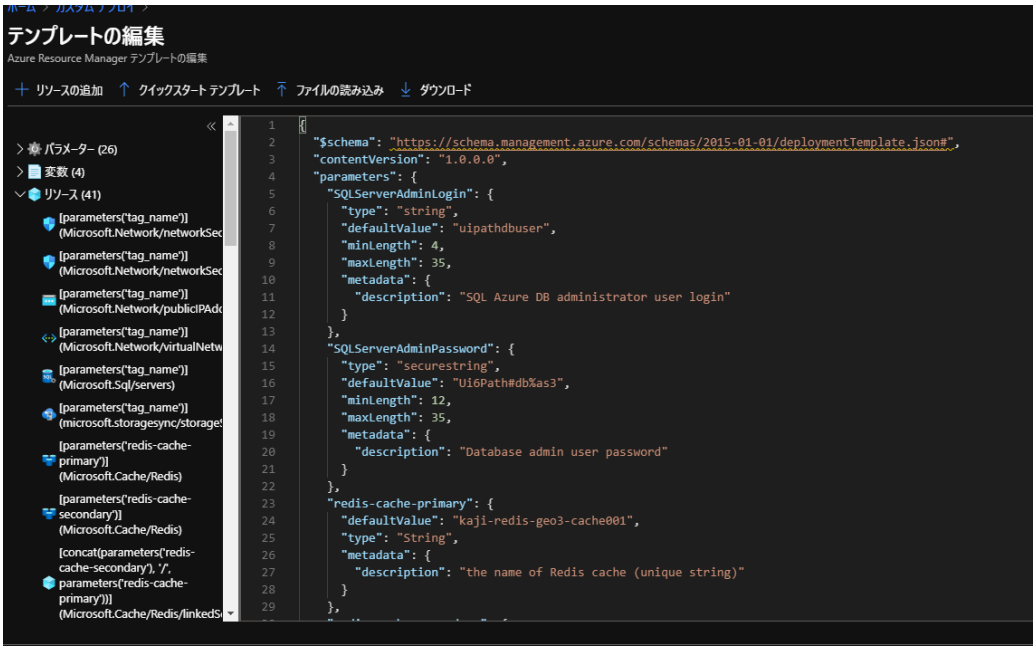
	Azure サービス	マネージド/アンマネージド	SKU / サイズ / プラン	冗長
Orchestrator	AzureVM	アンマネージド	Standard (Standard_D2s_v3)	ゾーン冗長
	Vnet / Subnet		-	-
	StandardLoadBalancer		Standard	ゾーン冗長
	NetworkSecurityGroup		-	-
	NatGateway		Standard	-
	PublicIP		Standard	-
	NetworkInterface		-	-
Cache	Azure Cache for Redis	マネージド	Premium 6GB	ローカル冗長
Database	SQL Server / SQL database	マネージド	DTU Premium P1 (125)	ゾーン冗長
Storage	StorageAccount	マネージド	Standard/Hot	ゾーン冗長
	AzureFileSync (2018.4 のみ)	マネージド	ファイル共有 64GB	ゾーン冗長
Elasticsearch	AzureVM	アンマネージド	Standard (Standard_D2s_v3)	ゾーン冗長
	Vnet / Subnet		-	-
	ApplicationGateway		Standard V2	ゾーン冗長
	NetworkSecurityGroup		-	-
	NetworkInterface		-	-

3. リソース作成

3.1. カスタムテンプレートのデプロイ

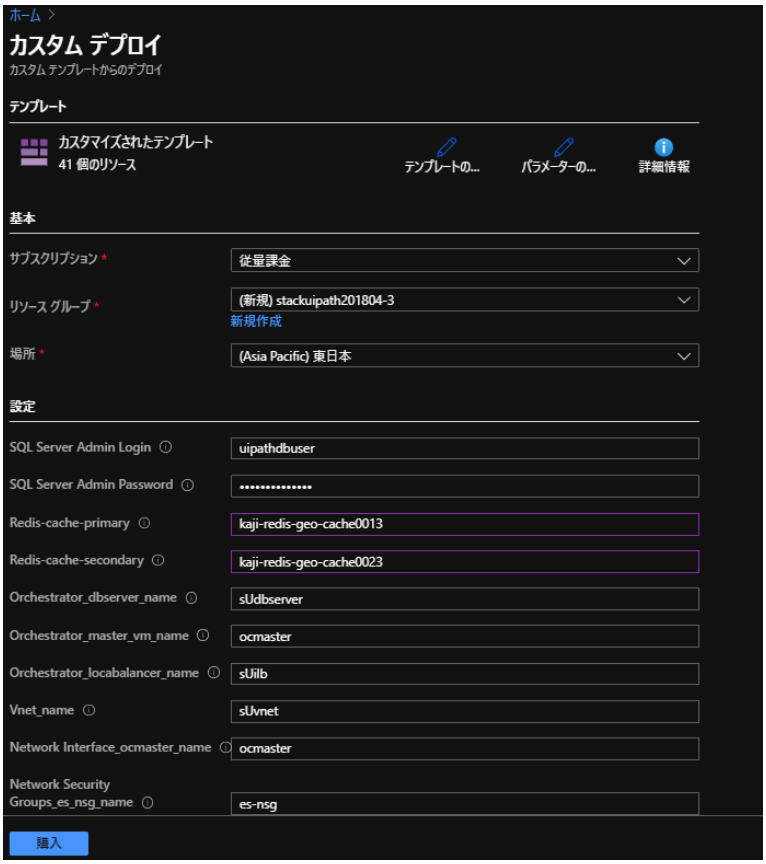
「エディターで独自のテンプレートを作成する」を選択

提供される ARM テンプレート(StackUiPath.json)をコピー＆ペースト



各設定値を入力し、購入をクリック

※クリックした時点でリソースが生成され Azure 内課金が発生します



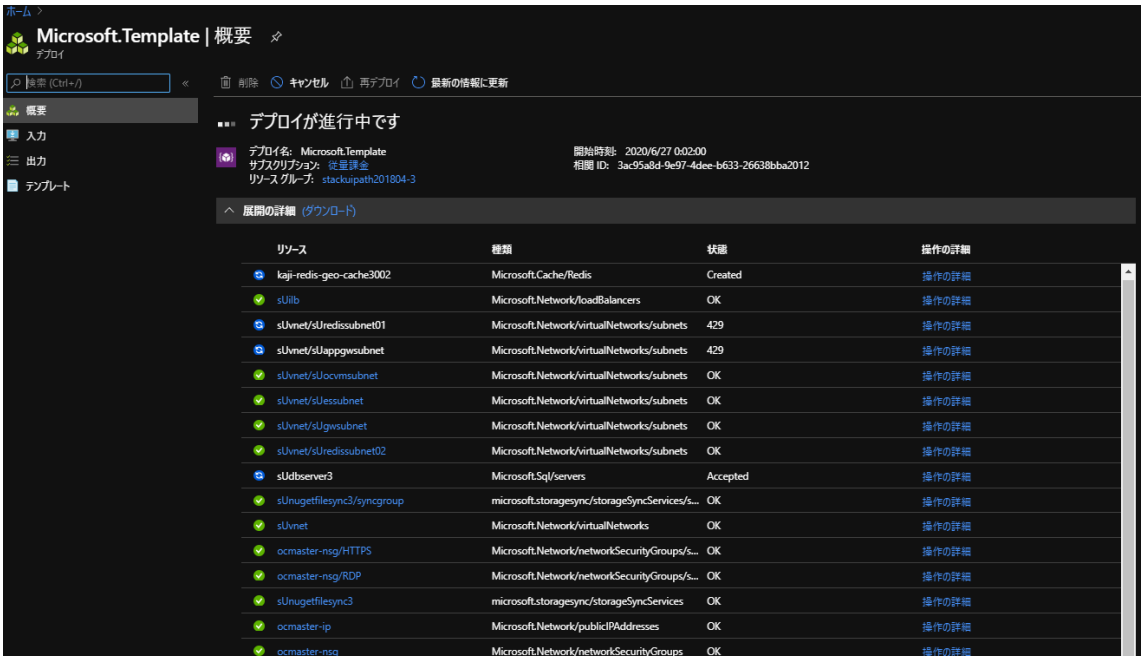
入力項目（ パラメータ ）一覧は下記の表を参照

パラメータ名	内容	デフォルト値	備考
Systemconfiguration	Redundancy（冗長構成）or Single（シングル構成）	Redundancy	※必須
Vnet_name	VNET の名称	sUvnet	※必須
Vnet_address_prefixes	VNET のアドレスプレフィックス	10.1.0.0/16	※必須
Vnet_gw_subnet_name	GW サブネット名称	suGwsubnet	
Vnet_gw_subnet_prefixes	GW サブネット用のアドレスプレフィックス	10.1.1.0/24	※必須
Vnet_ocvm_subnet_name	OrchestratorVM サブネット名称	sUocvmsubnet	
Vnet_ocvm_subnet_prefixes	OrchestratorVM サブネット用のアドレスプレフィックス	10.1.2.0/24	※必須
Vnet_redis_subnet_name	Redis サブネット名称	sUredissubnet	
Vnet_redis_subnet_prefixes	Redis サブネット用のアドレスプレフィックス	10.1.3.0/24	Single の場合不要
Vnet_es_subnet_name	Elasticsearch サブネット名称	sUessubnet	
Vnet_es_subnet_prefixes	Elasticsearch サブネット用のアドレスプレフィックス	10.1.4.0/24	Use Elasticsearch->False の場合不要
Vnet_appgw_subnet_name	ApplicationGateway サブネット名称	sUappgwsubnet	
Vnet_appgw_subnet_prefixes	ApplicationGateway サブネット用のアドレスプレフィックス	10.1.5.0/24	Use Elasticsearch->False の場合不要
Orchestrator_dbserver_name	SQL サーバ名	sUdbserver	※必須
SQL Server Admin Login	SQL サーバのログインユーザ名	uipathdbuser	※必須
SQL Server Admin Password	SQL サーバのログインパスワード	Ui6Path#db%as3(secureString)	※必須
Redis-cache	冗長構成を敷く場合の RedisCache	su-redis	Single の場合不要
Redis-cache-enable-nonssl	6379 ポートの nonSSL を有効	True	Single の場合不要
Orchestrator_locabalancer_name	Orchestrator のロードバランサ名	sUilb	※必須
Orchestrator_locabalancer_staticip	Orchestrator のロードバランサ staticIP	-	
Orchestrator_locabalancer_backendpool_name	Orchestrator のロードバランサ backendpool 名称	sUilbbepool	
Orchestrator_master_vm_name	Orchestrator のマスターイメージ用 VM	ocmaster	※必須
Network Interface_ocmaster_name	マスターVM のネットワークインタフェース名	ocmaster	※必須
Orchestrator_master_vm Publicip_name	マスターVM のパブリック IP	ocmaster-ip	※必須
Orchestrator_master_vm_nsg_name	マスターVM のネットワークセキュリティグループ名	ocmaster-nsg	※必須
Vm Windows OS Version	マスターVM 及び ES 用 VM の OS バージョン	WindowsServer 2019	
Orchestrator_master_vm Admin User Name	マスターVM の administrator ユーザ	-	※必須
Orchestrator_master_vm Admin Password	マスターVM の administrator パスワード	-	※必須
Use Elasticsearch	Elasticsearch を利用可否	True	Use Elasticsearch->False の場合不要
Network Security Groups_es_nsg_name	Elasticsearch のネットワークセキュリティグループ名	es-nsg	Use Elasticsearch-

			>False の場合不要
Elasticsearch_01_node_name	Elasticsearch01 ノード名	es01	Use Elasticsearch->False の場合不要
Elasticsearch_02_node_name	Elasticsearch02 ノード名	es02	Use Elasticsearch->False の場合不要
Elasticsearch_03_node_name	Elasticsearch03 ノード名	es03	Use Elasticsearch->False の場合不要
Elasticsearch_application Gateway_name	アプリケーションゲートウェイ名称	suappgw	Use Elasticsearch->False の場合不要
Elasticsearch_application Gateway_staticip	アプリケーションゲートウェイ staticIP	-	
App Gateway_Publicip_name	アプリケーションゲートウェイ publicIP 名称	Appgwpub-ip	
Network Security Groups_appgw_nsg_name	アプリケーションゲートウェイネットワークセキュリティグループ名	appgw-nsg	
My Source Ip	NSG 内で許可する GIP	-	
Storage Account_Name	ストレージアカウント名	-	※必須
Tag_name	タグ名	sTackUiPath	-
Orchestrator Admin Password	Orchestrator の admin 用パスワード Default / Host テナント用	-	-
Orch Passphrase	暗号化キー生成用のパスワード	Passw0rd!（secure String)	※必須
NatGateway_name	NatGateway の名称	suNatGw	※必須

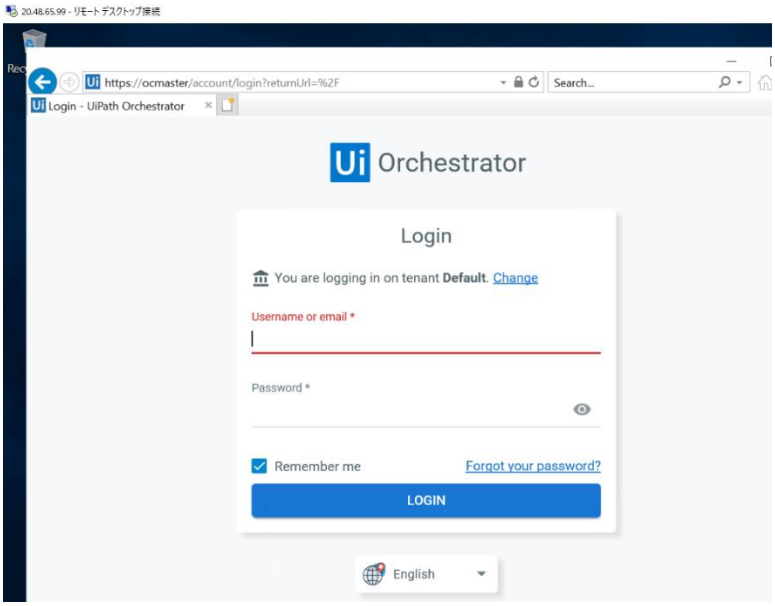
※作成される VM のインスタンスタイプ：Standard_D2s_v3

デプロイが進行しているのを確認



マスターVM に RDP でログイン

既に Orchestrator はインストールされているのでブラウザでログイン画面表示を確認及び、ログイン試行



※2018.4 の場合は、下記デフォルトの ID パスワードでログインを確認

Default テナント：

admin / 890iop

Host テナント：

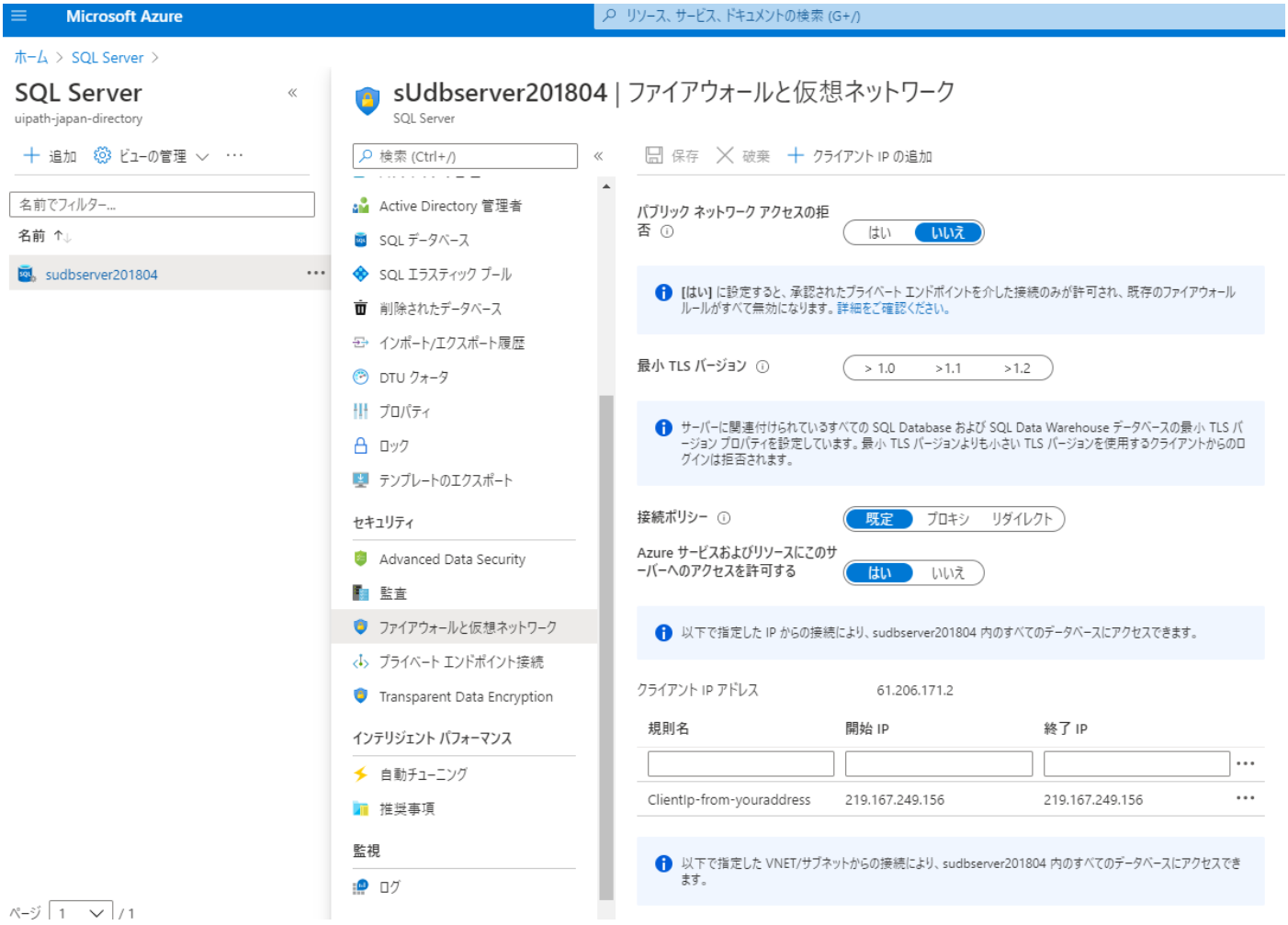
admin / 3edcVFR$

3.1. SQLServer アクセス制御

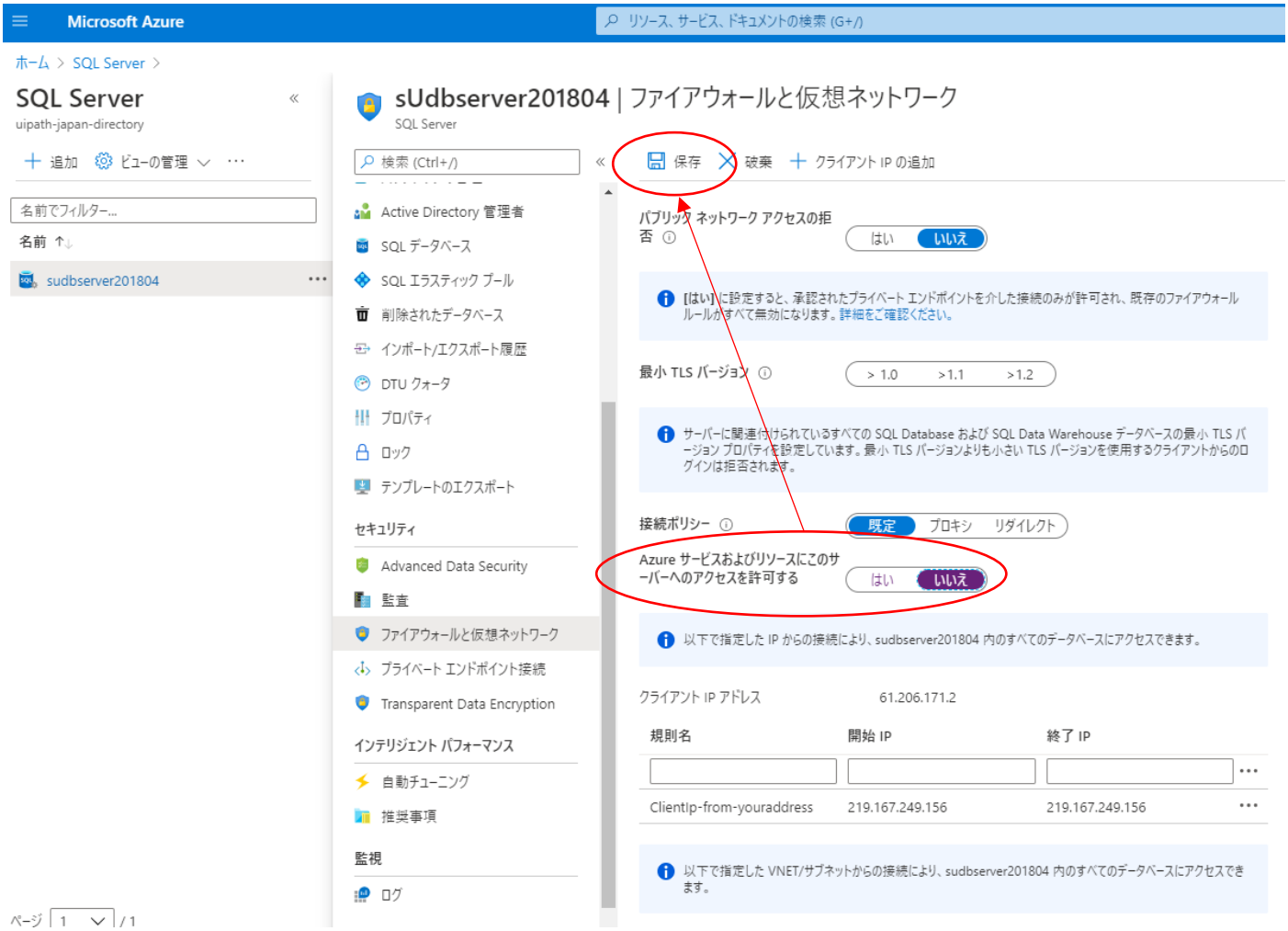
ARM テンプレートにて作成した SQLServer は全ての Azure 環境からのアクセスが許可された状態となっているため、アクセス制御設定を変更する。

Azure ポータルにて以下に遷移

ホーム ＞ SQL Server ＞ 対象 SQLServer ＞ファイアウォールと仮想ネットワーク



「Azure サービスおよびリソースにこのサーバーへのアクセスを許可する」を「いいえ」に変更し保存



4. Orchestrator 冗長化

4.1 Configure-platformNode.ps1

冗長構成を組む場合は下記の様に Configure-platformNode.ps1 を実行
本スクリプトについての詳細は下記を参照
https://docs.uipath.com/orchestrator/lang-ja/v2018.4/docs/configure-platformnodeps1-parameters
https://docs.uipath.com/orchestrator/lang-ja/v2019/docs/configure-platformnodeps1-parameters

スクリプト実行例：
［ 2018.4 ］

```
.¥Configure-PlatformNode.ps1 `
-mode ConfigurePrimary `
-websiteName "UiPath Orchestrator" `
-nugetPackagesPath D:¥Data¥NugetPackages `
-redisServer su201804.redis.cache.windows.net `
-redisPassword xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx `
-redisPort 6380
-outputCommandFile Install-Secondary.ps1
```

Web.Config 差分（ Configure-platformNode.ps1 実行前後 ）:

Before

```
~
    <add key="quartz.jobStore.clustered" value="false" />
~
    <add key="NuGet.Packages.Path" value="~/NuGetPackages" />
~
    <add key="NuGet.Activities.Path" value="~/NuGetPackages/Activities" />
~
    <add key="LoadBalancer.UseRedis" value="false" />
    <add key="LoadBalancer.Enabled" value="false" />
    <add key="LoadBalancer.Redis.ConnectionString" value="localhost:6379" />
~
    <sessionState mode="InProc" sqlConnectionString="" allowCustomSqlDatabase="true" />
~
```

After

```
~
    <add key="quartz.jobStore.clustered" value="true" />
~
    <add key="NuGet.Packages.Path" value="D:¥Data¥NugetPackages" />
~
    <add key="NuGet.Activities.Path" value="D:¥Data¥NugetPackages¥Activities¥" />
~
    <add key="LoadBalancer.UseRedis" value="true" />
    <add key="LoadBalancer.Enabled" value="false" />
    <add key="LoadBalancer.Redis.ConnectionString" value="kaji-redis-geo3-cache10001.redis.cache.windows.net:6380,password=FIlpUrmEhTRL8TIdFQTWaJrLPVeXKS1d5BigUMCzMSc=,ssl=True" />
~
    <sessionState mode="Custom" sqlConnectionString="" allowCustomSqlDatabase="true" customProvider="RedisStateStore">
      <providers>
        <add name="RedisStateStore" type="Microsoft.Web.Redis.RedisSessionStateProvider" connectionString="kaji-redis-geo3-cache10001.redis.cache.windows.net:6380,password=FIlpUrmEhTRL8TIdFQTWaJrLPVeXKS1d5BigUMCzMSc=,ssl=True" />
      </providers>
    </sessionState>
~
```

スクリプト実行例：

[2019.10]

```
.¥Configure-PlatformNode.ps1 `
-mode ConfigurePrimary `
-websiteName "UiPath Orchestrator" `
-redisServer su201910.redis.cache.windows.net `
-redisPassword xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx `
-redisPort 6380 `
-storageType Azure `
-storageLocation  "DefaultEndpointsProtocol=https;AccountName=usr;AccountKey
=...;EndpointSuffix=core.windows.net"
```

Web.Config 差分（ Configure-platformNode.ps1 実行前後 ）：

Before

```
~
    <add key="quartz.jobStore.clustered" value="false" />
~
    <add key="LoadBalancer.UseRedis" value="false" />
    <add key="LoadBalancer.Enabled" value="false" />
    <add key="LoadBalancer.Redis.ConnectionString" value="localhost:6379" />
~
    <add key="Storage.Type" value="FileSystem" />
    <add key="Storage.Location" value="RootPath=.¥Storage" />
~
```

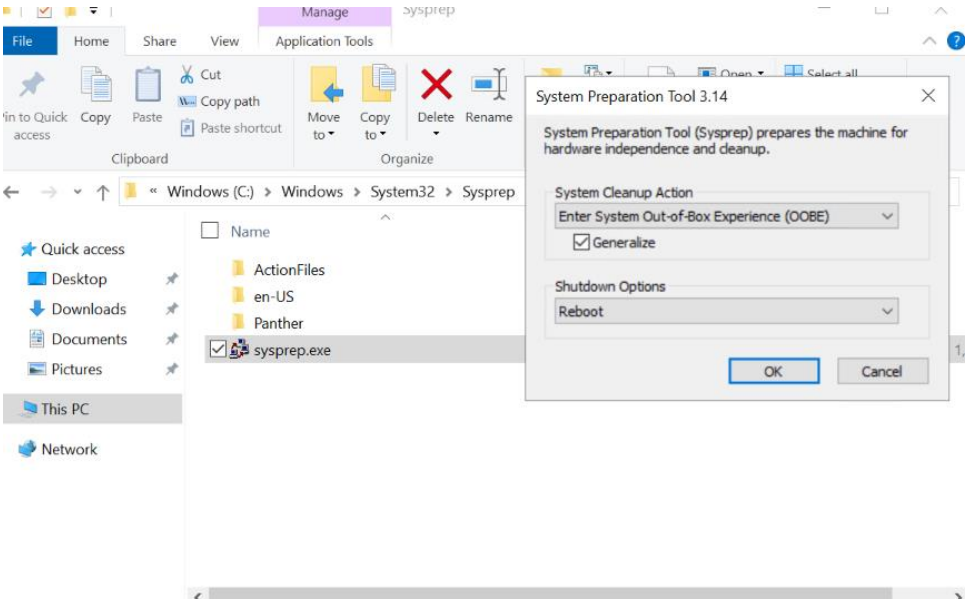
After

```
~
    <add key="quartz.jobStore.clustered" value="true" />
~
    <add key="LoadBalancer.UseRedis" value="true" />
    <add key="LoadBalancer.Enabled" value="false" />
    <add key="LoadBalancer.Redis.ConnectionString" value="kaji-redis-geo3-cache10001.redis.cache.windows.net:638
0,password=FIlpUrmEhTRL8TIdFQTWaJrLPVeXKS1d5BigUMCzMSc=,ssl=True" />
~
    <add key="Storage.Type" value="Azure" />
    <add key="Storage.Location" value="DefaultEndpointsProtocol=https;AccountName=usr;AccountKey=...;Endpoint
Suffix=core.windows.net" />
~
```

4.2 VM の複製

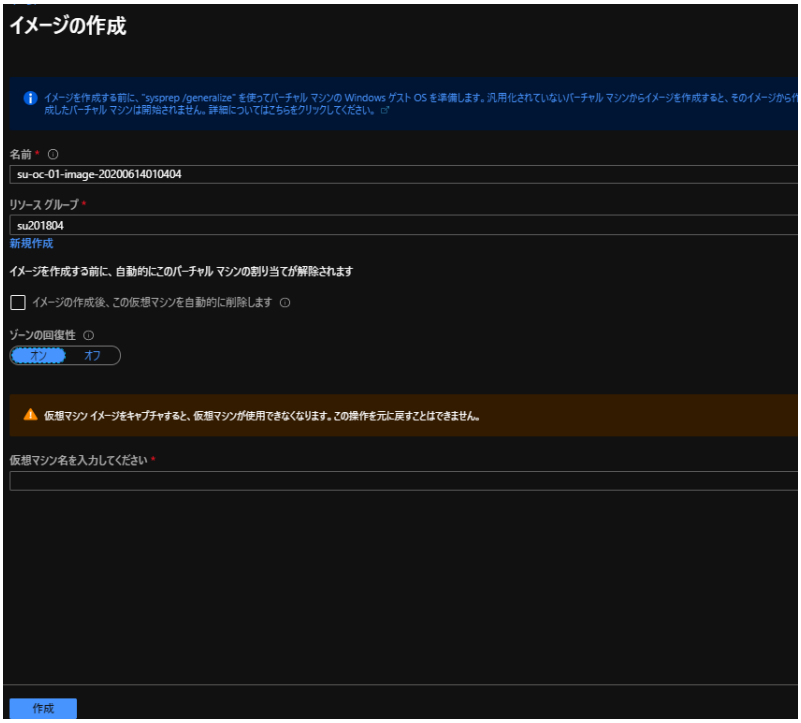
ここまで、マスターイメージ用の VM 内の作業でしたが、ここからはこの VM からマスターイメージを生成し、VM を複製できるように一般化(sysprep)を実施します。

Orchestrator Master VM 内で “ C:¥Windows¥System32¥Sysprep ” を実行

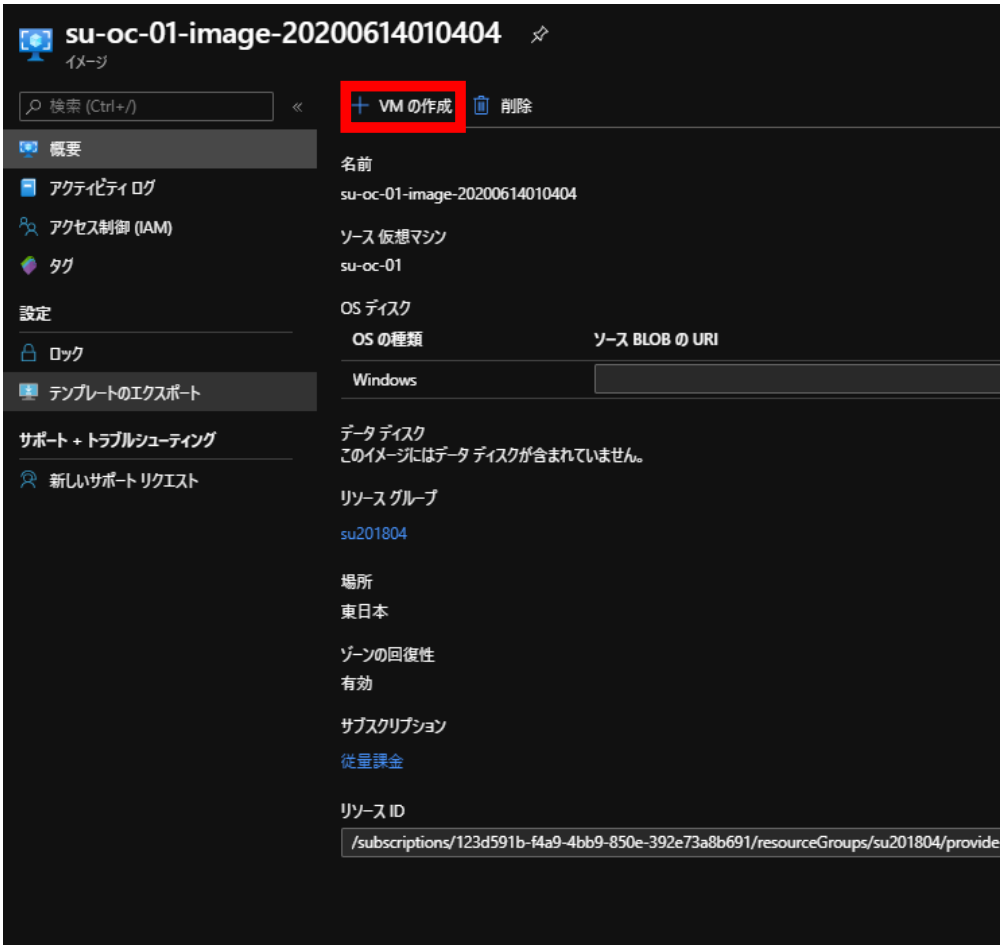


- System Cleanup Action -> **OOBE (プルダウン)**
- Generalize -> **有効 (チェックボックス)**
- Shutdown Options -> **Shutdown (プルダウン)**

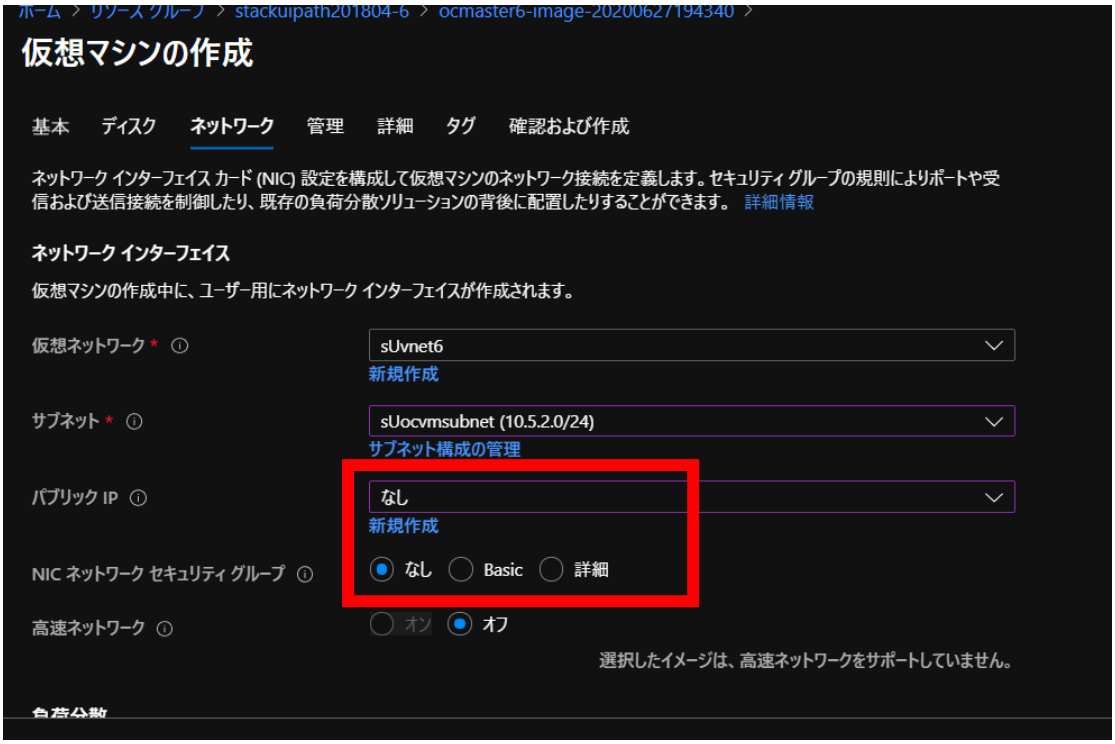
実行後、Master VM の RDP セッションが切れるので Azure ポータル上で
［ VM ］-> ［ キャプチャ ］を選択。下記のようにイメージの作成画面に遷移するのでマスターイメージ用 VM のマシン名を入力してイメージ作成を実施



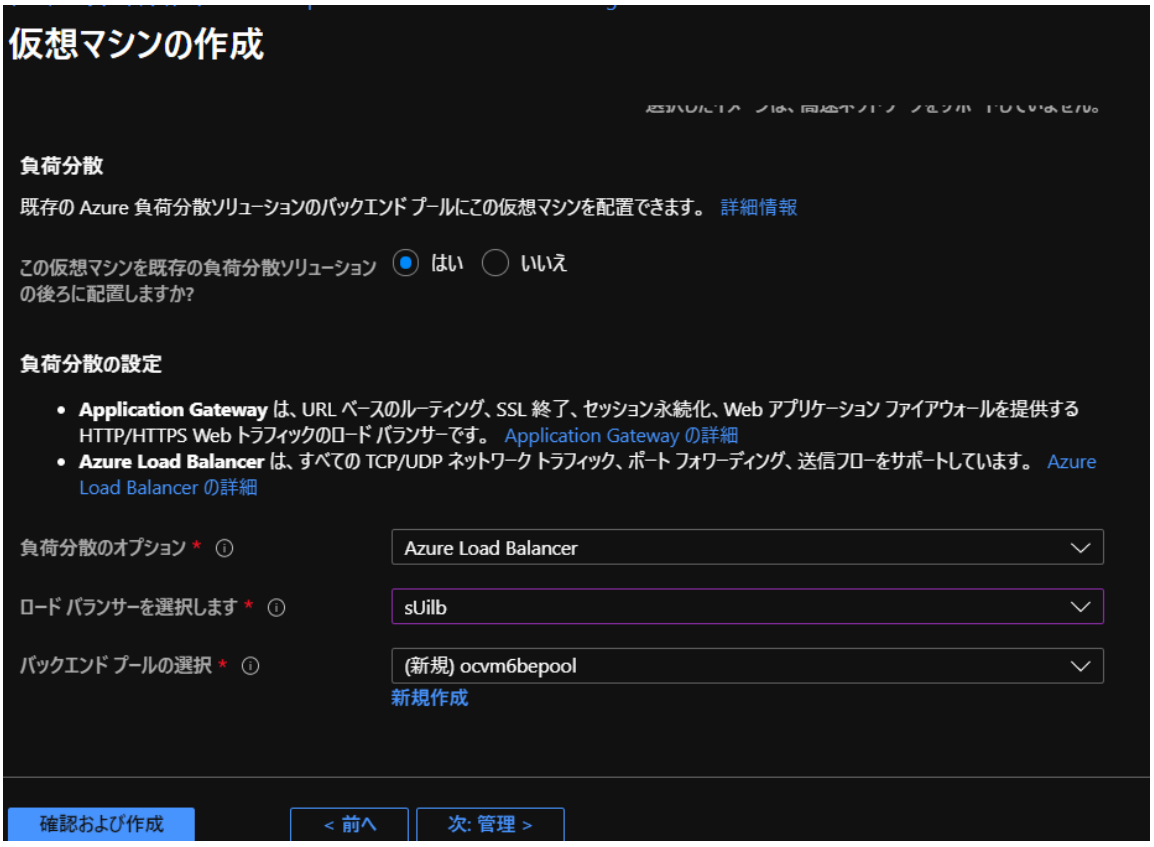
作成したイメージで必要な台数分だけ VM の作成



仮想マシン作成時に
パブリック IP 及び、NSG の生成しないように設定

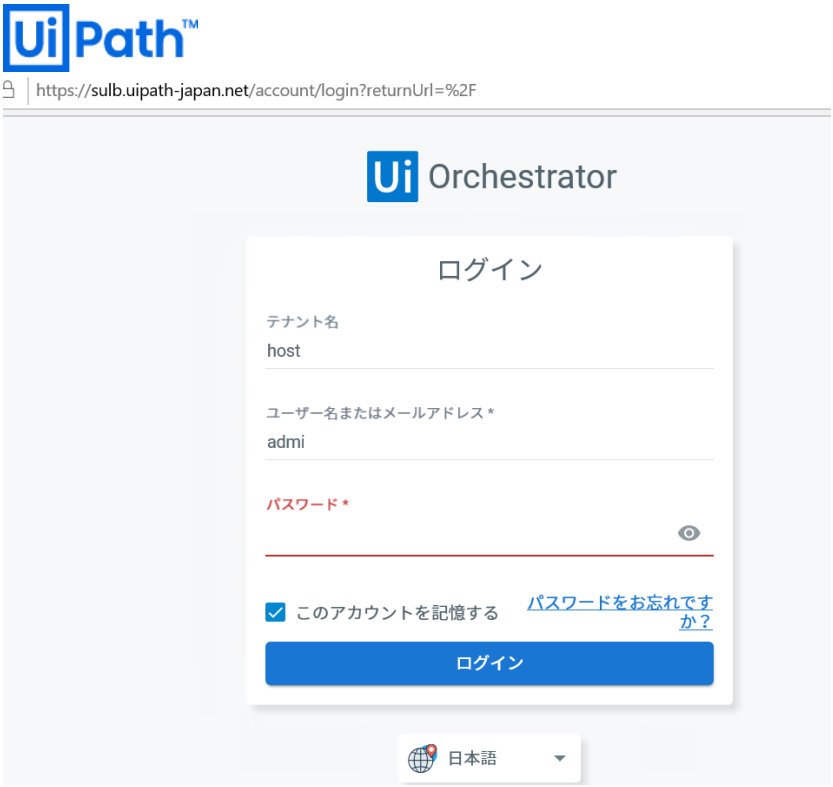


負荷分散の設定で指定の LB 及び、バックエンドプールを指定



以下のようにバックエンドプールに作成した VM を追加

LB 経由での簡易的にアクセス確認。
(この場合、SSL サーバ証明書をインストールし、ローカルの hosts に記載)

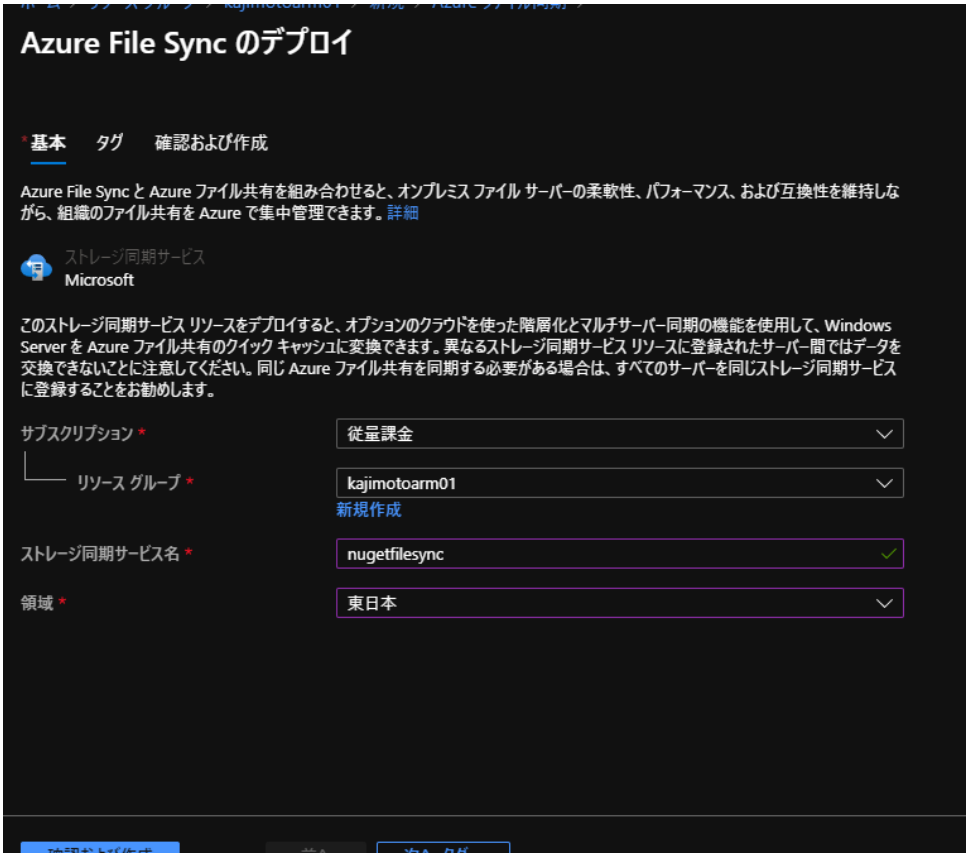


※[2018.4]の場合、以下 Azure ファイル同期作成の手順も実施してください。

冗長構成時の Nuget ファイル同期を実現するために、AzureFileSync で Nuget の同期させる

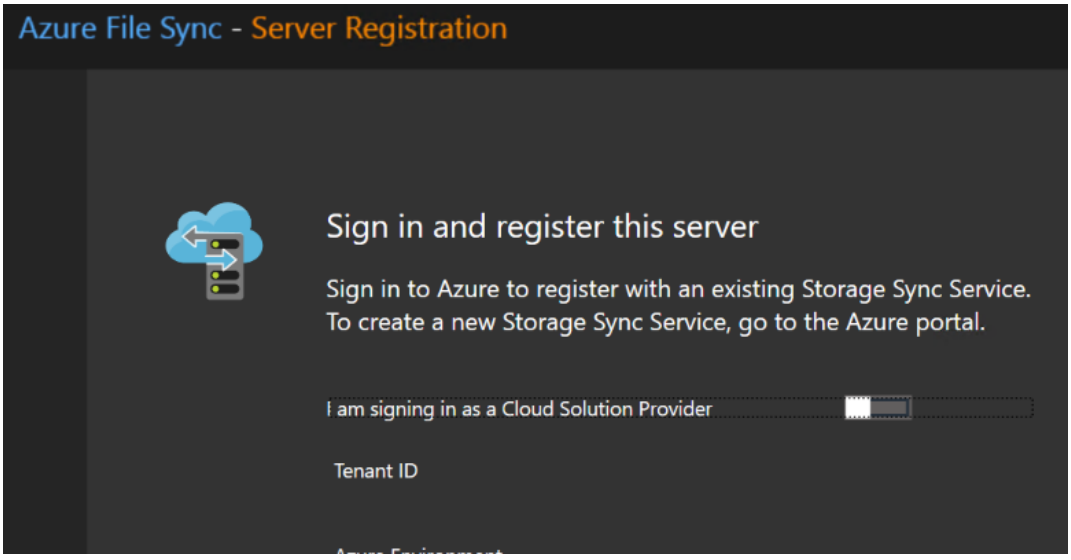
ファイル同期サービスを作成

ここでストレージ同期サービス名称は一意



次にエージェントをインストール（下記リンクの手順に従う)
https://docs.microsoft.com/ja-jp/azure/storage/files/storage-sync-files-server-registration#install-the-azure-file-sync-agent

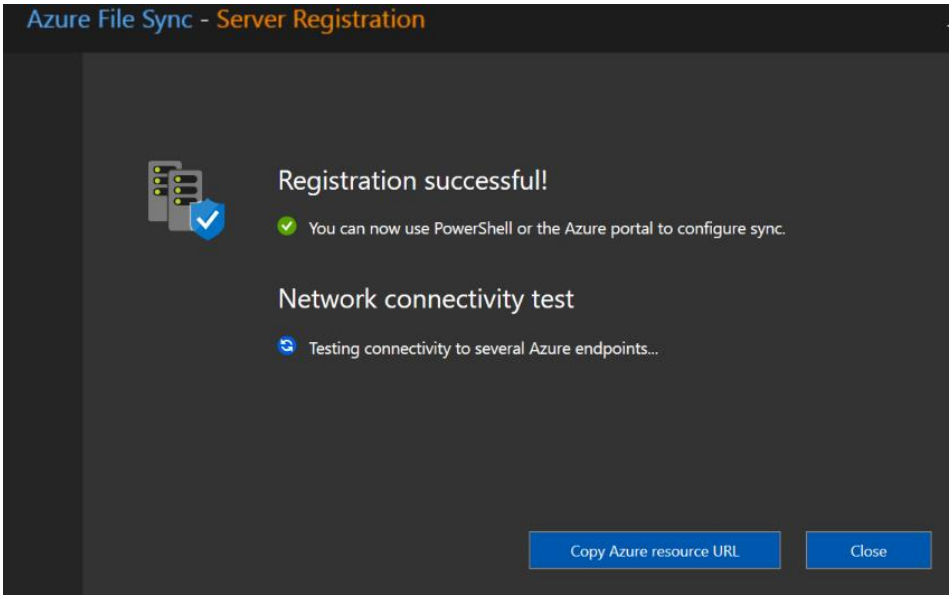
インストール後、下記のように登録



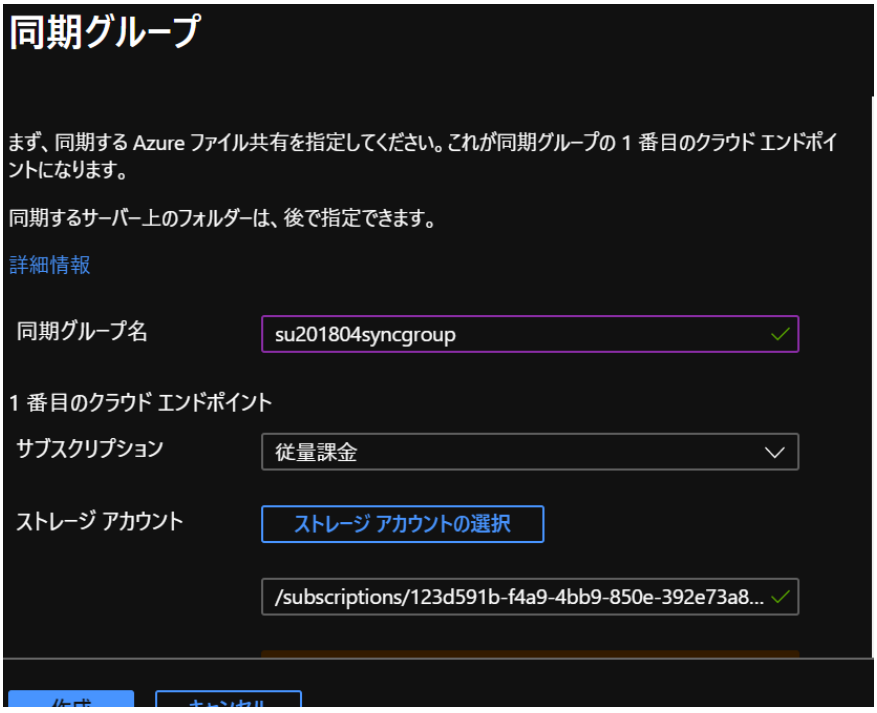
下記のように登録


Azure File Sync - Server Registration
Choose a Storage Sync Service
Azure Subscription
従量課金
Subscription ID: 123d591b-f4a9-4bb9-850e-392e73a8b691
Resource Group
su201804
Storage Sync Service
nugetfilesync
Register

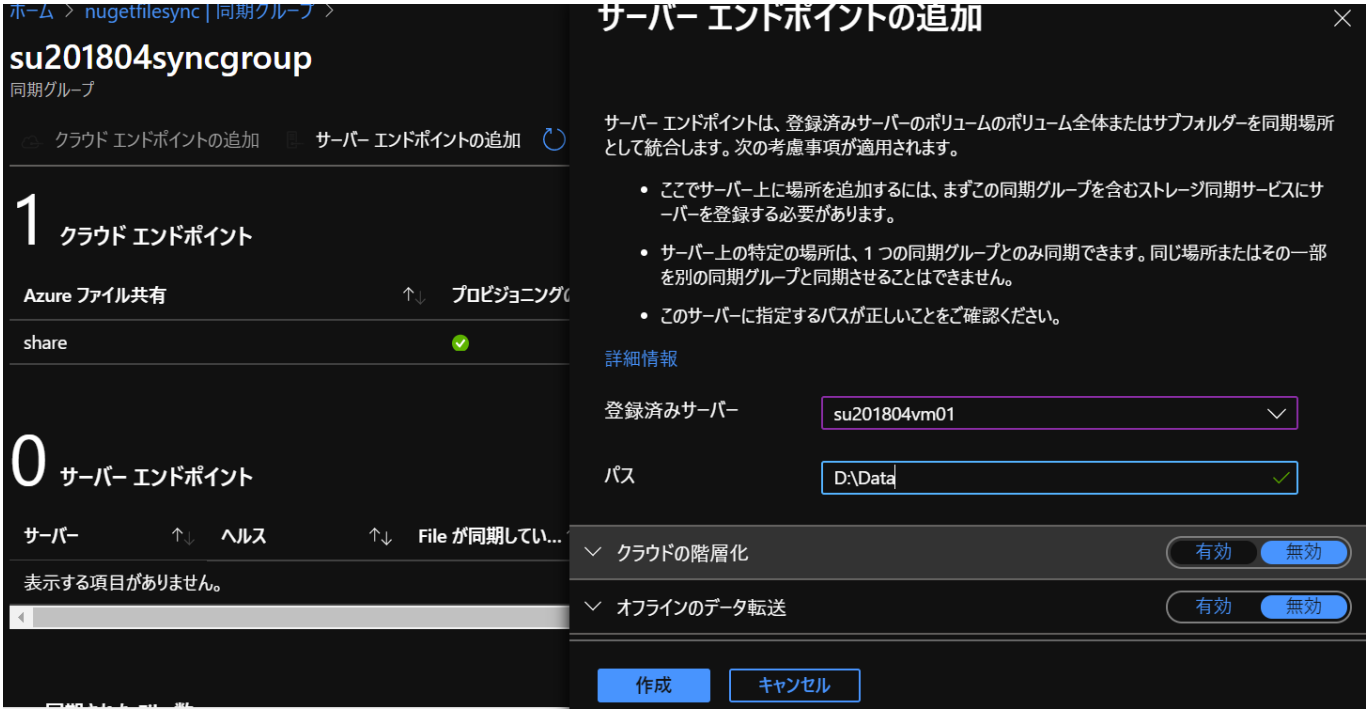
接続確認



同期グループを作成



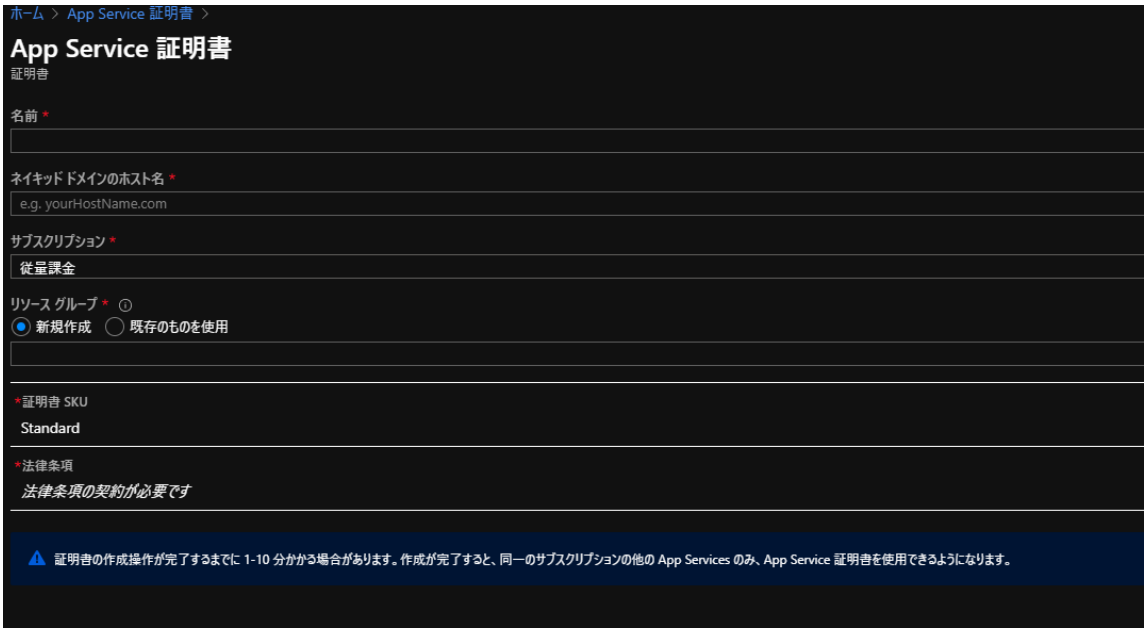
サーバエンドポイントの追加



-------2018.4 での Nuget のファイル同期作業はここまで----------

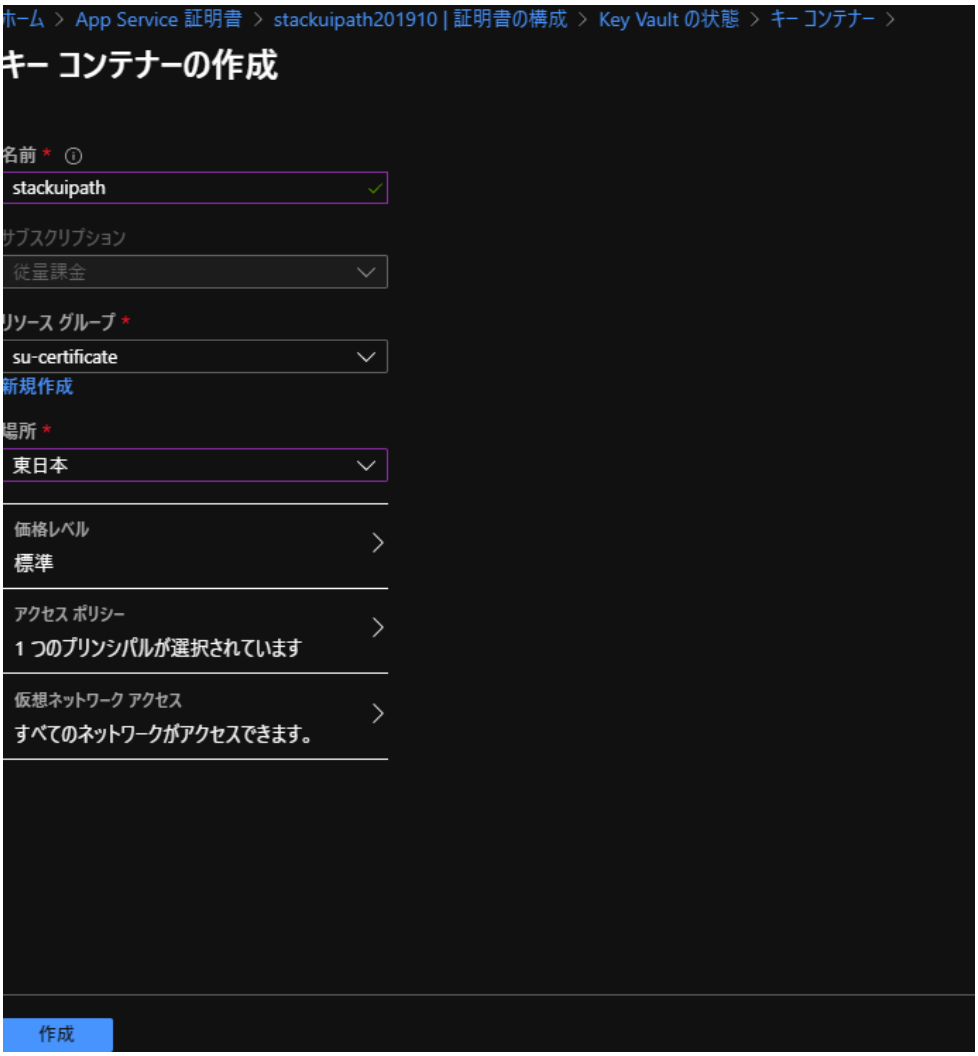
4.3 [オプション] Azure 内での SSL サーバ証明書の準備

AppService 証明書を用意



証明書の構成で手順 1 を実施

キーコンテナーを作成



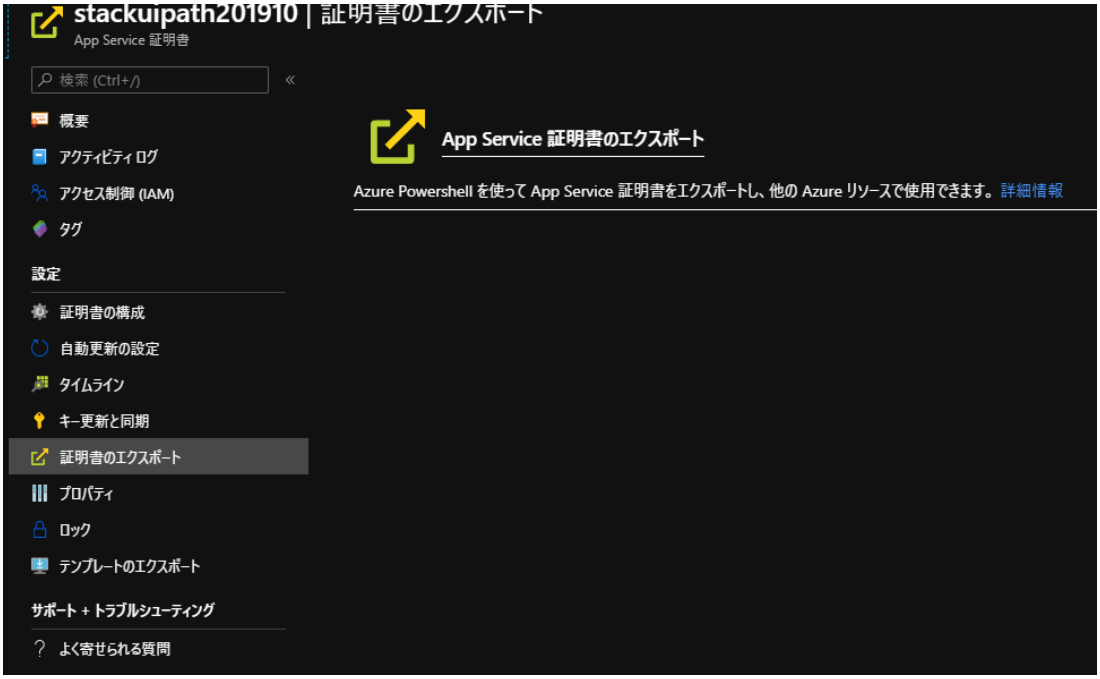
作成した証明書は下記ブログを参考に VM にエクスポート

https://azure.github.io/AppService/2017/02/24/Creating-a-local-PFX-copy-of-App-Service-Certificate.html

手順２を実施

※ドメインはあらかじめ取得済み

証明書のエクスポート



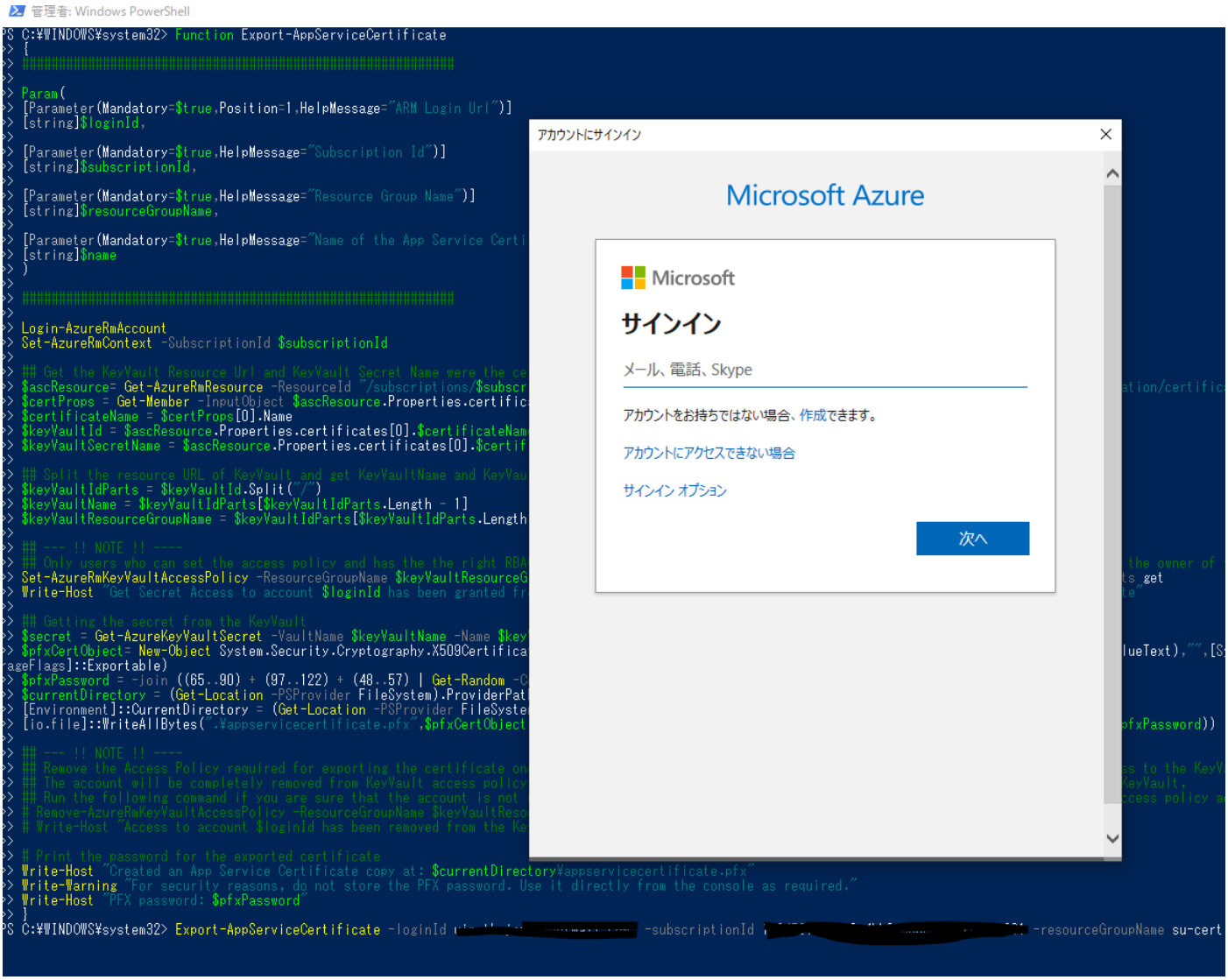
下記のコマンドで関数を生成

```
Function Export-AppServiceCertificate
{
#############################################################

Param(
[Parameter(Mandatory=$true,Position=1,HelpMessage="ARM Login Url")]
[string]$loginId,

[Parameter(Mandatory=$true,HelpMessage="Subscription Id")]
[string]$subscriptionId,

[Parameter(Mandatory=$true,HelpMessage="Resource Group Name")]
[string]$resourceGroupName,

[Parameter(Mandatory=$true,HelpMessage="Name of the App Service Certificate Resource")]
[string]$name
)

#############################################################

Login-AzureRmAccount
Set-AzureRmContext -SubscriptionId $subscriptionId

## Get the KeyVault Resource Url and KeyVault Secret Name were the certificate is stored
$ascResource= Get-AzureRmResource -ResourceId
"/subscriptions/$subscriptionId/resourceGroups/$resourceGroupName/providers/Microsoft.CertificateRegistration/certificateOrd
ers/$name"
$certProps = Get-Member -InputObject $ascResource.Properties.certificates[0] -MemberType NoteProperty
$certificateName = $certProps[0].Name
$keyVaultId = $ascResource.Properties.certificates[0].$certificateName.KeyVaultId
$keyVaultSecretName = $ascResource.Properties.certificates[0].$certificateName.KeyVaultSecretName

## Split the resource URL of KeyVault and get KeyVaultName and KeyVaultResourceGroupName
$keyVaultIdParts = $keyVaultId.Split("/")
$keyVaultName = $keyVaultIdParts[$keyVaultIdParts.Length - 1]
$keyVaultResourceGroupName = $keyVaultIdParts[$keyVaultIdParts.Length - 5]

## --- !! NOTE !! ----
## Only users who can set the access policy and has the the right RBAC permissions can set the access policy on KeyVault, if
the command fails contact the owner of the KeyVault
Set-AzureRmKeyVaultAccessPolicy -ResourceGroupName $keyVaultResourceGroupName -VaultName $keyVaultName -UserPrincipalName
$loginId -PermissionsToSecrets get
Write-Host "Get Secret Access to account $loginId has been granted from the KeyVault, please check and remove the policy
after exporting the certificate"

## Getting the secret from the KeyVault
$secret = Get-AzureKeyVaultSecret -VaultName $keyVaultName -Name $keyVaultSecretName
$pfxCertObject= New-Object System.Security.Cryptography.X509Certificates.X509Certificate2 -
ArgumentList @([Convert]::FromBase64String($secret.SecretValueText),"",[System.Security.Cryptography.X509Certificates.X509Ke
yStorageFlags]::Exportable)
$pfxPassword = -join ((65..90) + (97..122) + (48..57) | Get-Random -Count 50 | % {[char]$_})
$currentDirectory = (Get-Location -PSProvider FileSystem).ProviderPath
[Environment]::CurrentDirectory = (Get-Location -PSProvider FileSystem).ProviderPath
[io.file]::WriteAllBytes(".¥appservicecertificate.pfx",$pfxCertObject.Export([System.Security.Cryptography.X509Certificates.
X509ContentType]::Pkcs12,$pfxPassword))

## --- !! NOTE !! ----
## Remove the Access Policy required for exporting the certificate once you have exported the certificate to prevent giving
the account prolonged access to the KeyVault
## The account will be completely removed from KeyVault access policy and will prevent to account from accessing any
keys/secrets/certificates on the KeyVault,
## Run the following command if you are sure that the account is not used for any other access on the KeyVault or login to
the portal and change the access policy accordingly.
# Remove-AzureRmKeyVaultAccessPolicy -ResourceGroupName $keyVaultResourceGroupName -VaultName $keyVaultName -
UserPrincipalName $loginId
# Write-Host "Access to account $loginId has been removed from the KeyVault"

# Print the password for the exported certificate
Write-Host "Created an App Service Certificate copy at: $currentDirectory¥appservicecertificate.pfx"
Write-Warning "For security reasons, do not store the PFX password. Use it directly from the console as required."
Write-Host "PFX password: $pfxPassword"
}
```

さらにその関数を利用して下記のようにコマンド実行

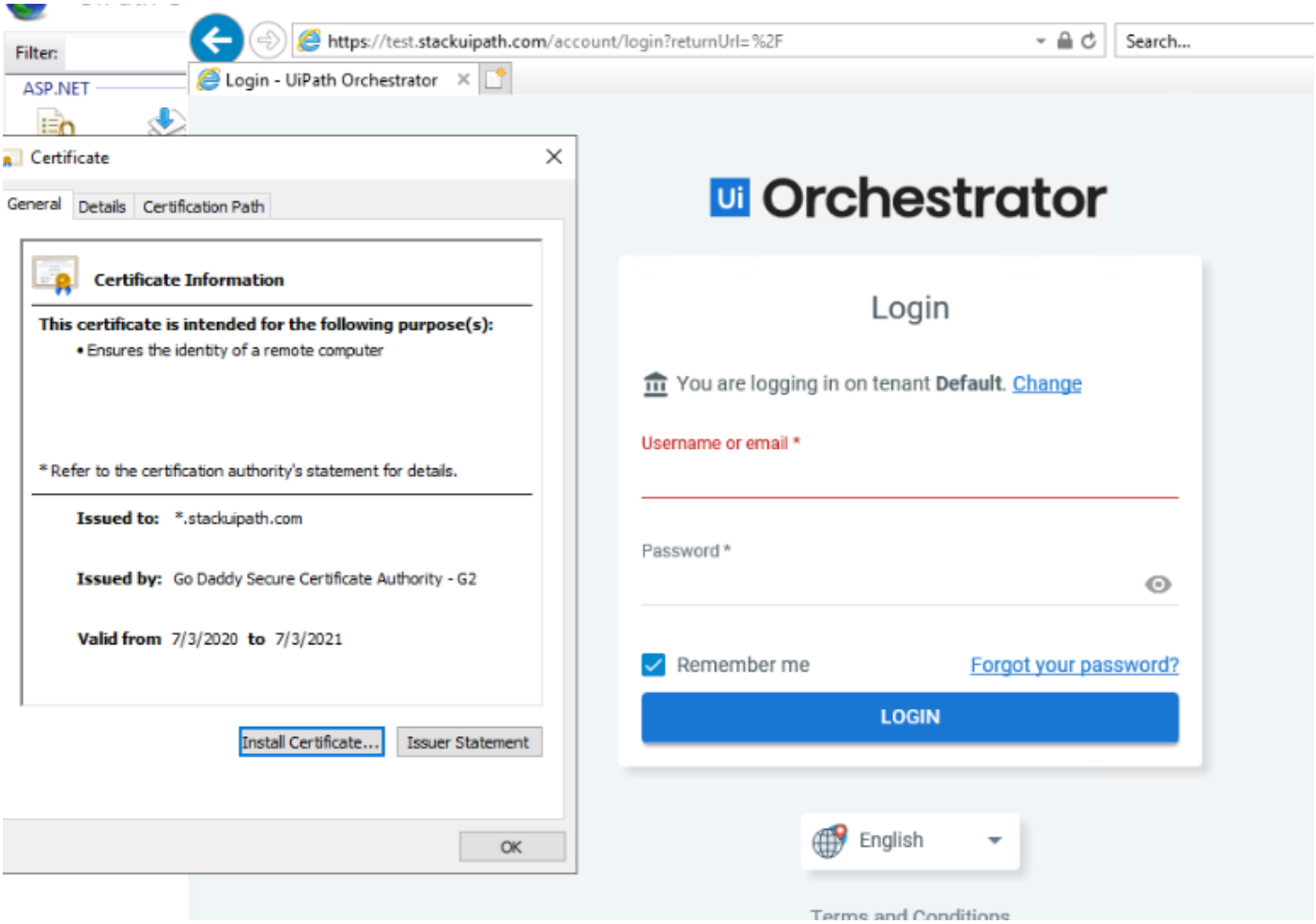
```
Export-AppServiceCertificate -loginId yourarmemail@domain.com -subscriptionId yoursubid
-resourceGroupName resourceGroupNameOfYourAppServiceCertificate -name
appServiceCertificateName
```

実行例：



```
Get Secret Access to account uipath.japan@hotmail.com has been granted from the KeyVault, please check and
remove the policy after exporting the certificate
Created an App Service Certificate copy at: C:¥WINDOWS¥system32¥appservicecertificate.pfx
警告: For security reasons, do not store the PFX password. Use it directly from the console as required.
PFX password: sxlJT6IQjWhF9aSUzCLqedXwHfK73piM5orEn4BGA18gNbv2tu


PS C:¥WINDOWS¥system32>
```

作成された PFX をインポートし、動作確認



5. Elasticsearch (ES) Deployment 手順

※カスタムテンプレートのパラメータ “Use Elasticsearch” を True にして VM 及び Application Gateway の作成を事前しておく事。

1. ARM テンプレートでデプロイした Azure Vnet 内の ES subnet にデプロイされている ES 用の VM 3 台にログインできる事を確認 (ex.) es01, es02, es03)

2. 踏み台サーバから作成した 3 台に RDP ログインを行い、以下 Firewall 設定 Powershell コマンドを実行し、Ping 疎通・Elasticsearch/Kibana 用に Port を開放、許可
   - Set-NetFirewallRule -DisplayName "File and Printer Sharing (Echo Request - ICMPv4-*)" -Enabled True -Profile Public, Private, Domain
   - New-NetFirewallRule -DisplayName "ElasticKibana" -Direction Inbound -Action Allow -EdgeTraversalPolicy Block -Protocol TCP -LocalPort 5601,9200-9300 -Profile Any -Enabled True

3. 以下のインストーラ・ファイルをダウンロード
   - Google Chrome installer
   - Elasticsearch MSI installer (v7.6.2)
   - Kibana exe (v7.6.2)
   - NSSM (v2.24) (Kibana を Windows Service 化するツール)

   ※2018.4 の場合はこちら

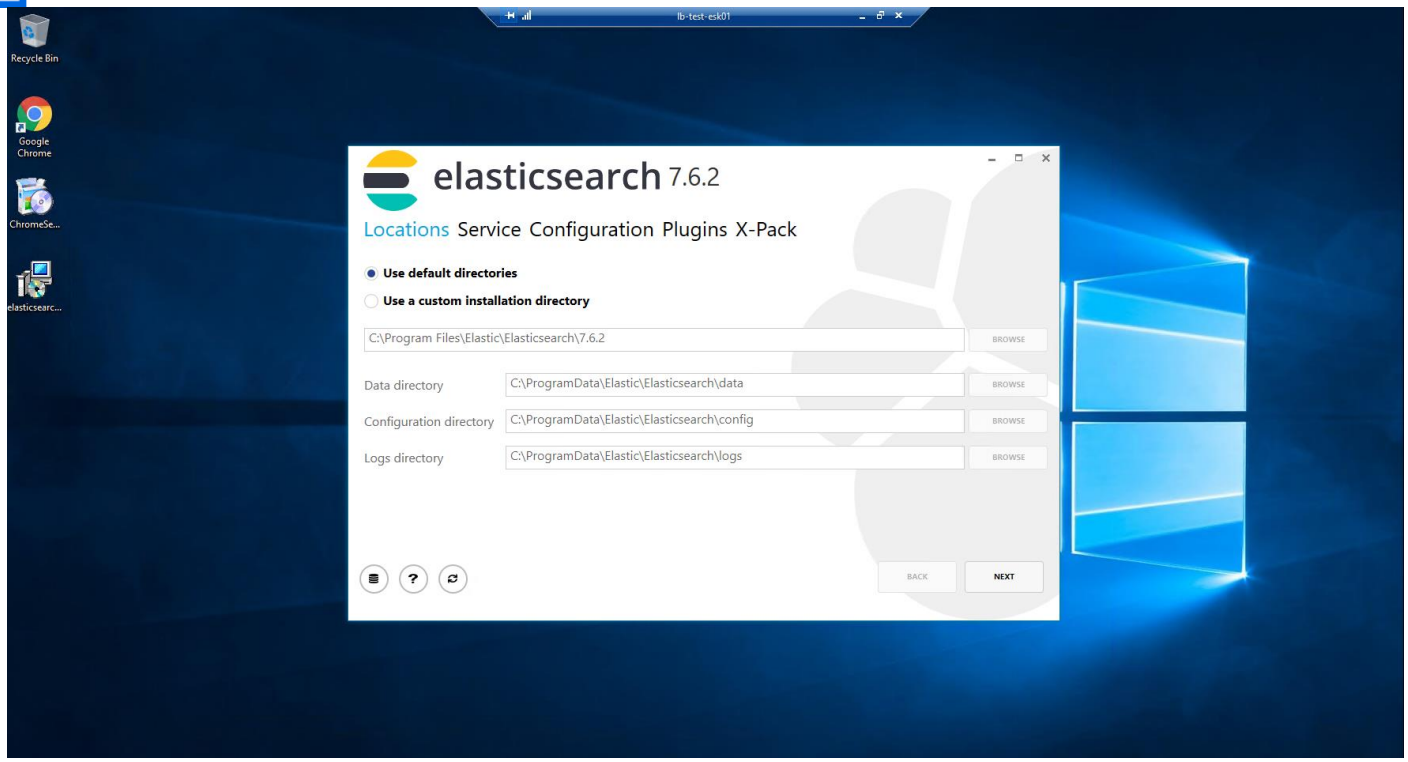
   ES: https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.8.10.msi

   Kibana: https://artifacts.elastic.co/downloads/kibana/kibana-6.8.10-windows-x86_64.zip

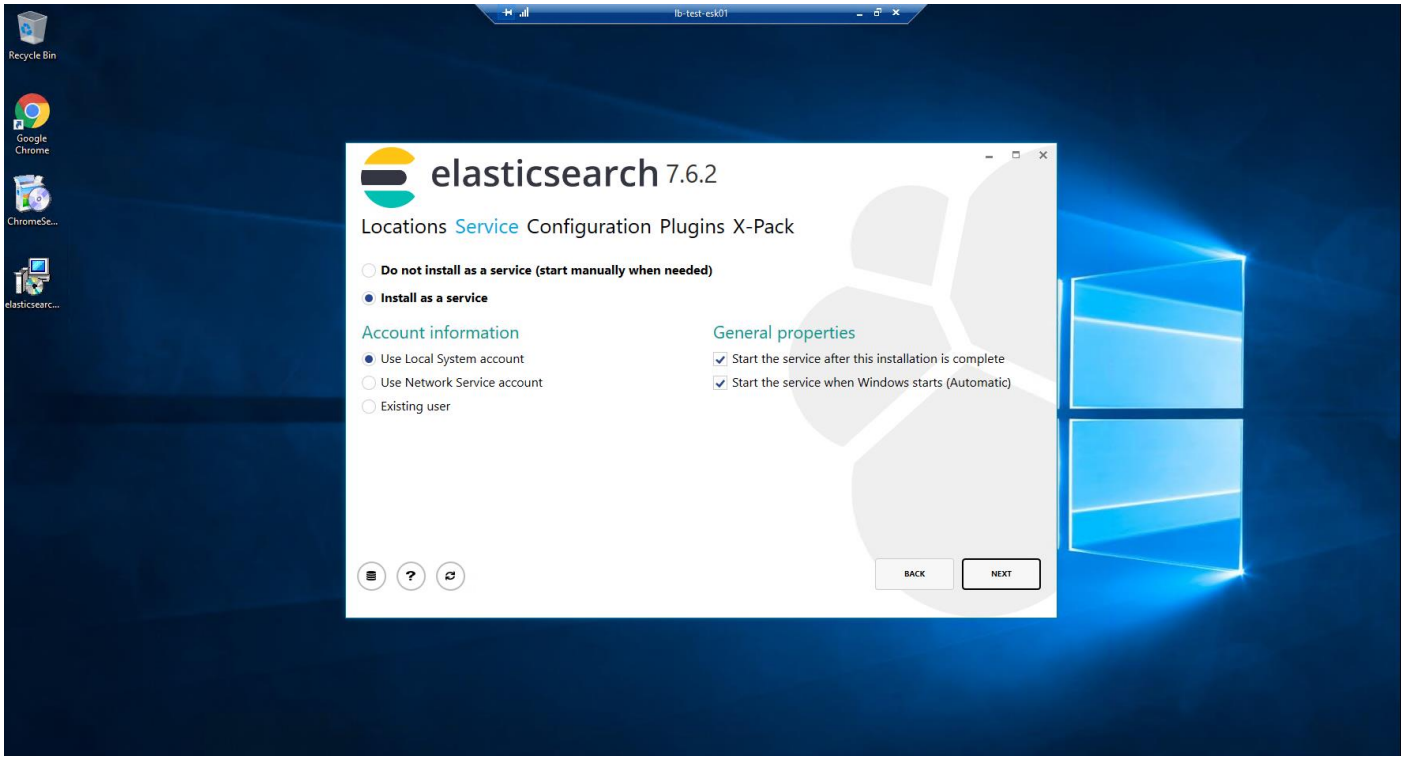
   また、Java のインストールが必要になってくるので

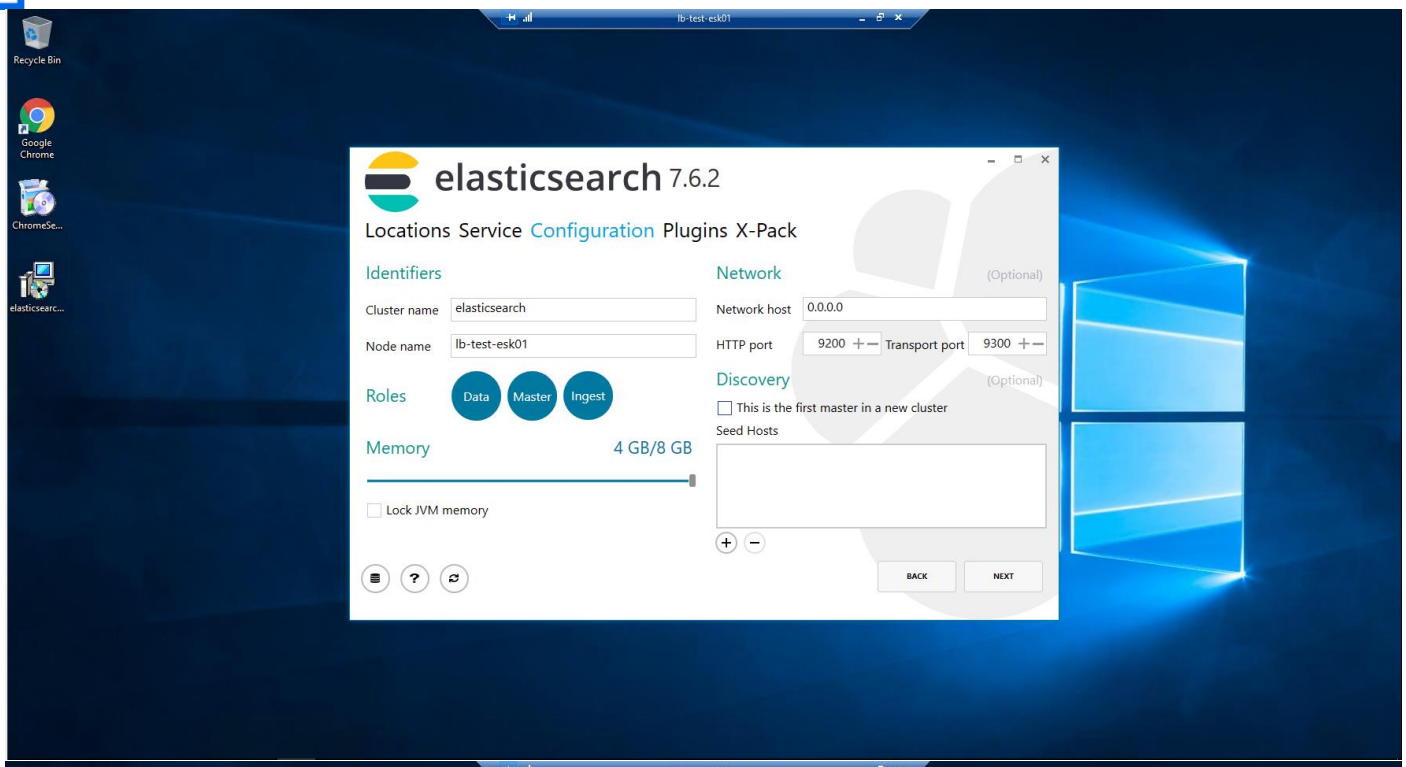
   https://www.java.com/en/download/manual.jsp からダウンロードします。

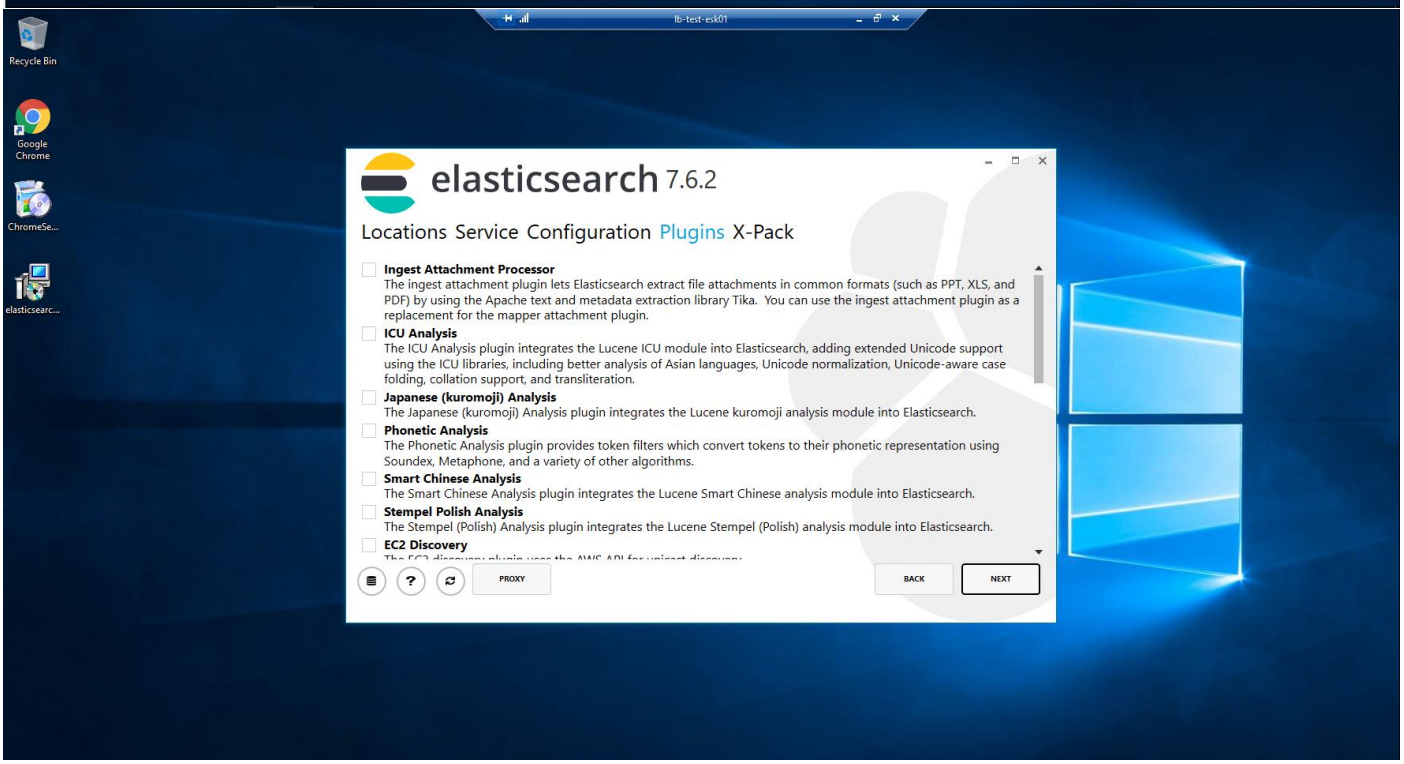
4. 作成した Elasticsearch 用 VM 全てのノードで手順(3)でダウンロードした Chrome をインストール
5. 作成した Elasticsearch 用 VM 全てのノードで手順(3)でダウンロードした Elasticsearch msi を実行し、インストール

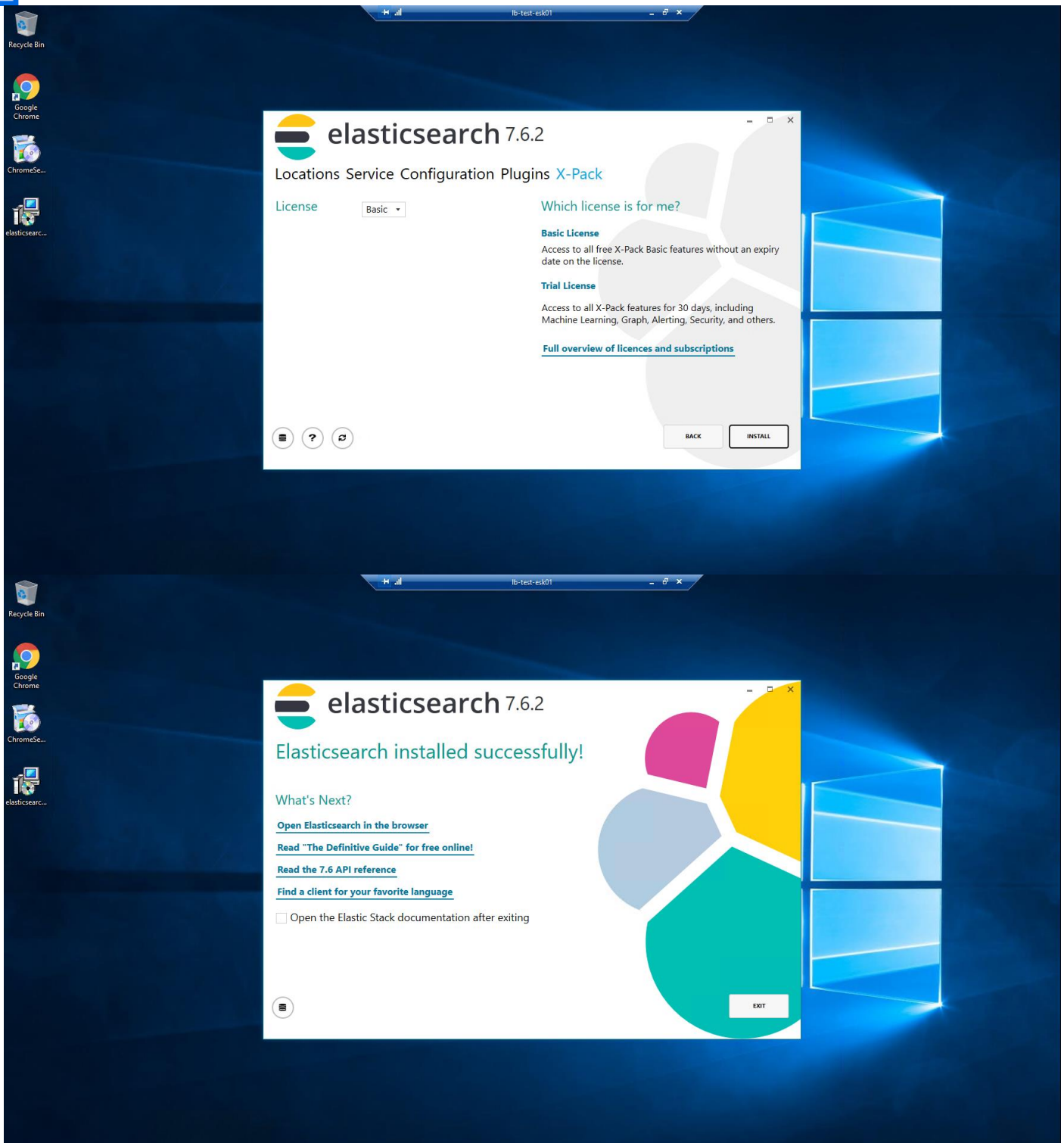


インストールディレクトリやログ、データディレクトリを変更したい場合はこの画面で変更しておくこと

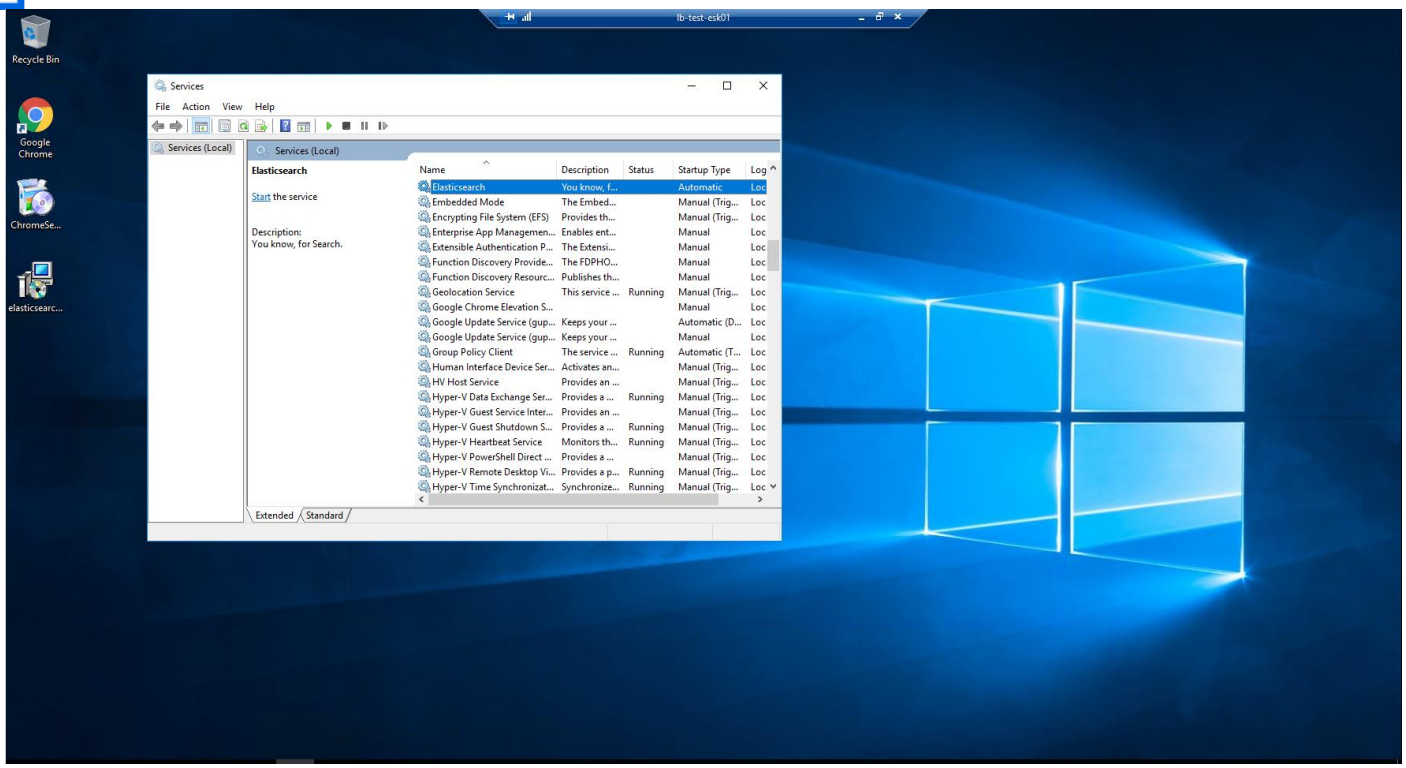



lb-test-esk01
Recycle Bin
Google Chrome
ChromeSe...
elasticsearc...
elasticsearch 7.6.2
Locations Service Configuration Plugins X-Pack
Identifiers
Cluster name elasticsearch
Node name lb-test-esk01
Roles
Data
Master
Ingest
Memory 4 GB/8 GB
Lock JVM memory
Network (Optional)
Network host 0.0.0.0
HTTP port 9200
Transport port 9300
Discovery (Optional)
This is the first master in a new cluster
Seed Hosts
BACK
NEXT


lb-test-esk01
Recycle Bin
Google Chrome
ChromeSe...
elasticsearc...
elasticsearch 7.6.2
Locations Service Configuration Plugins X-Pack
Ingest Attachment Processor
The ingest attachment plugin lets Elasticsearch extract file attachments in common formats (such as PPT, XLS, and PDF) by using the Apache text and metadata extraction library Tika. You can use the ingest attachment plugin as a replacement for the mapper attachment plugin.
ICU Analysis
The ICU Analysis plugin integrates the Lucene ICU module into Elasticsearch, adding extended Unicode support using the ICU libraries, including better analysis of Asian languages, Unicode normalization, Unicode-aware case folding, collation support, and transliteration.
Japanese (kuromoji) Analysis
The Japanese (kuromoji) Analysis plugin integrates the Lucene kuromoji analysis module into Elasticsearch.
Phonetic Analysis
The Phonetic Analysis plugin provides token filters which convert tokens to their phonetic representation using Soundex, Metaphone, and a variety of other algorithms.
Smart Chinese Analysis
The Smart Chinese Analysis plugin integrates the Lucene Smart Chinese analysis module into Elasticsearch.
Stempel Polish Analysis
The Stempel (Polish) Analysis plugin integrates the Lucene Stempel (Polish) analysis module into Elasticsearch.
EC2 Discovery
PROXY
BACK
NEXT



インストールが終了したら上記画面のような Window が表示されるので、Exit を押下。



Elasticsearch Service が登録されていることを確認

6. Elasticsearch をインストールした全てのノードで elasticsearch.yml を開き、*discovery.need_hosts* と *cluster.initial_master_nodes* セクションを追加し、elasticserch.yml 内にある<>のパラメータを設定

elasticsearch.yml

```
bootstrap.memory_lock: false
cluster.name: <任意の Cluster 名>
http.port: 9200
network.host: 0.0.0.0
node.data: true
node.ingest: true
node.master: true
node.max_local_storage_nodes: 1
node.name: <ノードの hostname>
path.data: C:¥ProgramData¥Elastic¥Elasticsearch¥data
path.logs: C:¥ProgramData¥Elastic¥Elasticsearch¥logs
transport.tcp.port: 9300
xpack.license.self_generated.type: basic
xpack.security.enabled: false
discovery.seed_hosts: [
    "<ノード 1 の hostname>",
    "<ノード 2 の hostname>",
    "<ノード 3 の hostname>"]
cluster.initial_master_nodes: [
    "<ノード 1 の hostname>",
    "<ノード 2 の hostname>",
    "<ノード 3 の hostname>"]
```

2018.4(ESv6.8)の場合、

```
bootstrap.memory_lock: false
cluster.name: <任意の Cluster 名>
http.port: 9200
network.host: 0.0.0.0
node.data: true
node.ingest: true
node.master: true
node.max_local_storage_nodes: 1
node.name: <ノードの hostname>
path.data: C:¥ProgramData¥Elastic¥Elasticsearch¥data
path.logs: C:¥ProgramData¥Elastic¥Elasticsearch¥logs
transport.tcp.port: 9300
xpack.license.self_generated.type: basic
xpack.security.enabled: false
discovery.zen.ping.unicast.hosts:
   - <ノード 1 の hostname>
   - <ノード 2 の hostname>
   - <ノード 3 の hostname>
discovery.zen.minimum_master_nodes: 2
```

参照: https://www.elastic.co/guide/en/elasticsearch/reference/6.8/discovery-settings.html

7. 動作確認のため、各ノードで Elasticsearch サービスを再起動後、Elasticsearch Cluster が正常に組まれていることを Chrome で以下 URL ごとの記載観点から確認

   - http://<いずれかのノードの hostname>:9200/

     - ✓ cluster_uuid が「_na_」ではなく、UUID が付与されていること(例: -tDeUcYLQ6GHFBUi0iRvhQ)

     - ✓ 「_na_」になっている場合は、Firewall を無効化する

   - http://<いずれかのノードの Hostname>:9200/_cluster/health?pretty

     - ✓ cluster_name が指定された elasticsearch.yml で指定した cluster_name になっていること
     - ✓ status が green であること
     - ✓ number_of_master_nodes が指定した Master node の数と合っていること(この場合は 3)

   - http:// <いずれかのノードの Hostname>:9200/_cat/nodes?v&s=name

     - ✓ 3 台の Nodes の情報が表示されること

8. Kibana のインストール(「UiPath Orchestrator 構築 ステップバイステップガイド」を参照)
   https://www.uipath.com/ja/resources/whitepaper/orchestrator-install-guide
   Kibana インストール後、kibana.yml 内の設定値を以下のようにコメントオフを外し、変更

```
server.port: 5601
server.host: 0.0.0.0
elasticsearch.hosts: ["http://es01:9200", "http://es02:9200", "http://es03:9200" ]
```

9. (Optional) Elasticsearch X-Pack Security 機能有効化
   参考: https://www.elastic.co/guide/en/elasticsearch/reference/7.6/configuring-tls.html
   (1) Elasticsearch 全ノードで%ES_PATH_CONF%配下に certs フォルダを作成し、公開鍵・秘密鍵を配置(本手順書では.pem 形式のものを配置)
   (2) 全ノードの elasticsearch.yml を赤文字部分を追加・変更
   elasticsearch.yml

```
bootstrap.memory_lock: false
cluster.name: <任意の Cluster 名>
http.port: 9200
network.host: 0.0.0.0
node.data: true
node.ingest: true
node.master: true
node.max_local_storage_nodes: 1
```

```
node.name: <ノードの hostname>
path.data: C:¥ProgramData¥Elastic¥Elasticsearch¥data
path.logs: C:¥ProgramData¥Elastic¥Elasticsearch¥logs
transport.tcp.port: 9300
xpack.license.self_generated.type: basic
xpack.security.enabled: true
discovery.seed_hosts: [
    "<ノード 1 の hostname>",
    "<ノード 2 の hostname>",
    "<ノード 3 の hostname>"]
cluster.initial_master_nodes: [
    "<ノード 1 の hostname>",
    "<ノード 2 の hostname>",
    "<ノード 3 の hostname>"]
xpack.security.transport.ssl.enabled: true
xpack.security.transport.ssl.verification_mode: certificate
xpack.security.transport.ssl.key: <上記で配置した秘密鍵のファイル名を絶対パスと共に記載※>
xpack.security.transport.ssl.certificate: <上記で配置した公開鍵のファイル名を絶対パスと共に記載>
```

※例:
xpack.security.tranposrt.ssl.key: C:¥ProgramData¥Elastic¥Elasticsearch¥config¥certs¥uipath-japan-net-secretkey.pem

(3) 全ノードで elasticsearch サービスを再起動

(4) いずれかのノードで C:¥Program Files¥Elastic¥Elasticsearch¥7.6.2¥bin に CMD で cd し、以下コマンドを実行(自動で Built-in ユーザのパスワードを生成、メモを控えること)

```
elasticsearch-setup-passwords auto
```

当コマンドにより、自動で Elasticsearch の Built-in ユーザのパスワードが生成されコンソール出力されるため、必ず控えること。

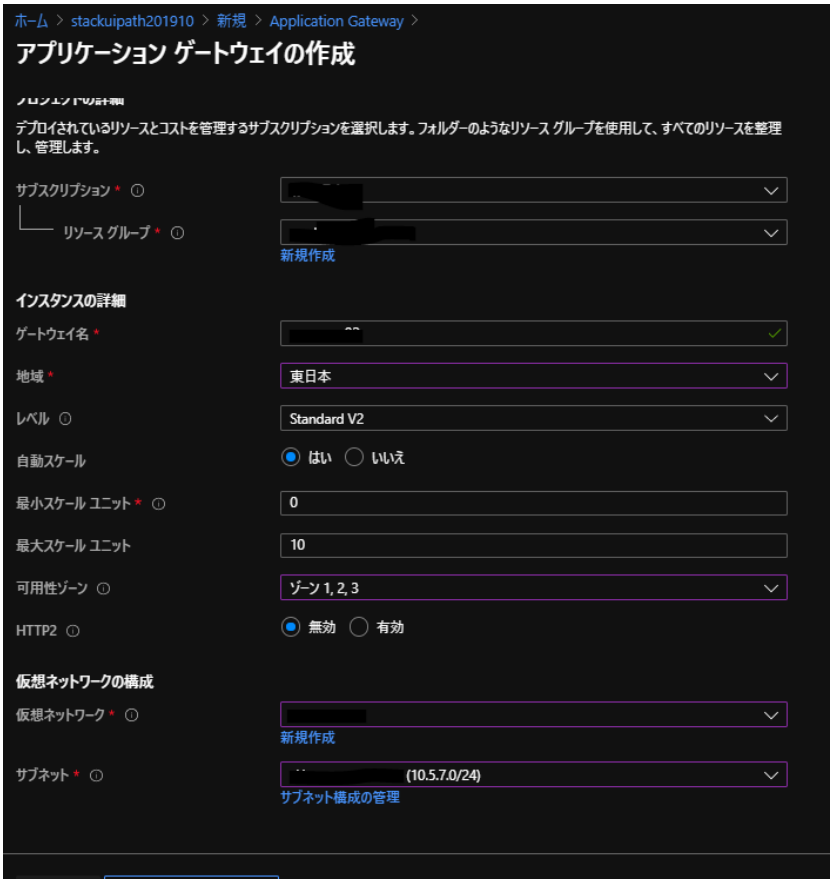
(5) 全ノードの kibana.yml の以下該当箇所に elasticsearch-setup-passwords によって生成された kibana ユーザ名とパスワードを記載

```
elasticsearch.username: "kibana"

elasticsearch.password: "<password>"
```

その後、kibana サービスを再起動

10.Azure Application Gateway (AGW)の Deploy・設定

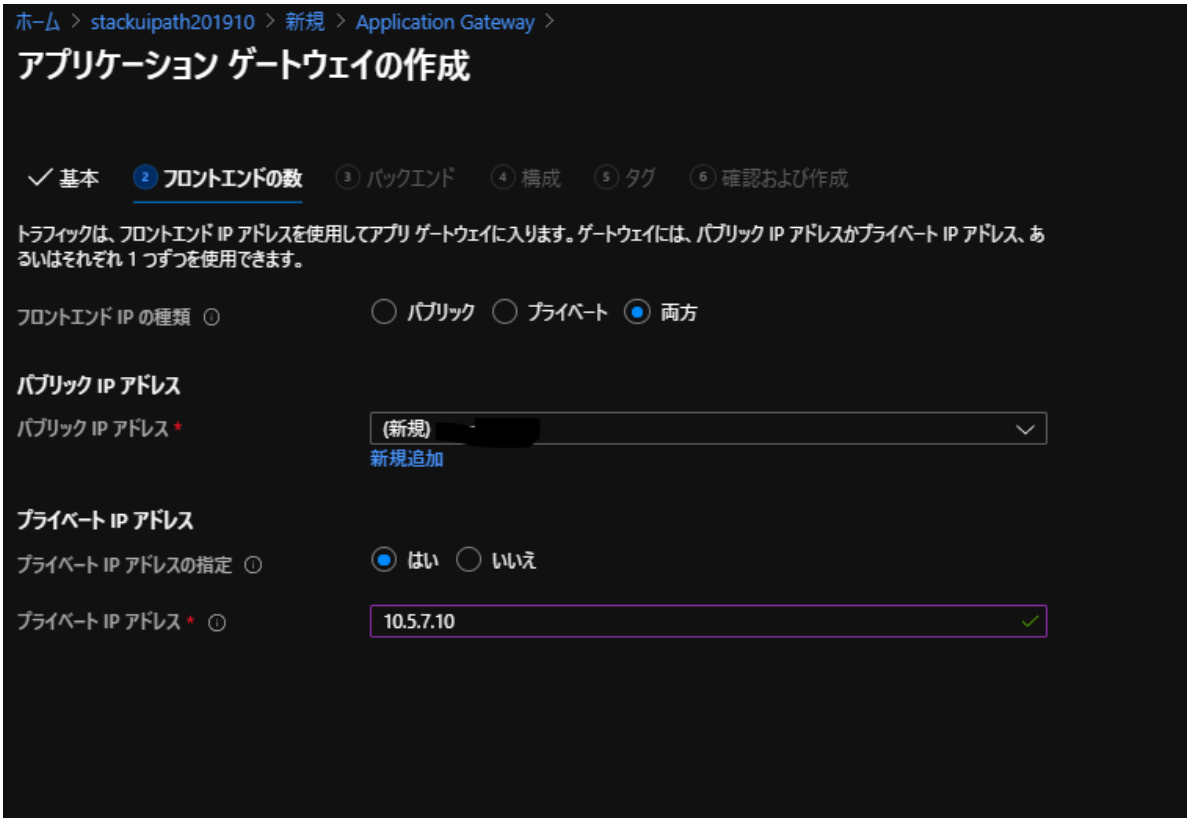
※カスタムテンプレートのパラメータ “Use Elasticsearch” を Ture にして Application Gateway の作成を事前しておくと、以下 ApplicationGateway の設定手順を省けます。手動で設定する場合、参考にして下さい。



・レベル：Standard V2

・自動スケール有効、最小：0、最大：10

・ゾーン冗長を実現させるために可用性ゾーンは 1, 2 を選択する。



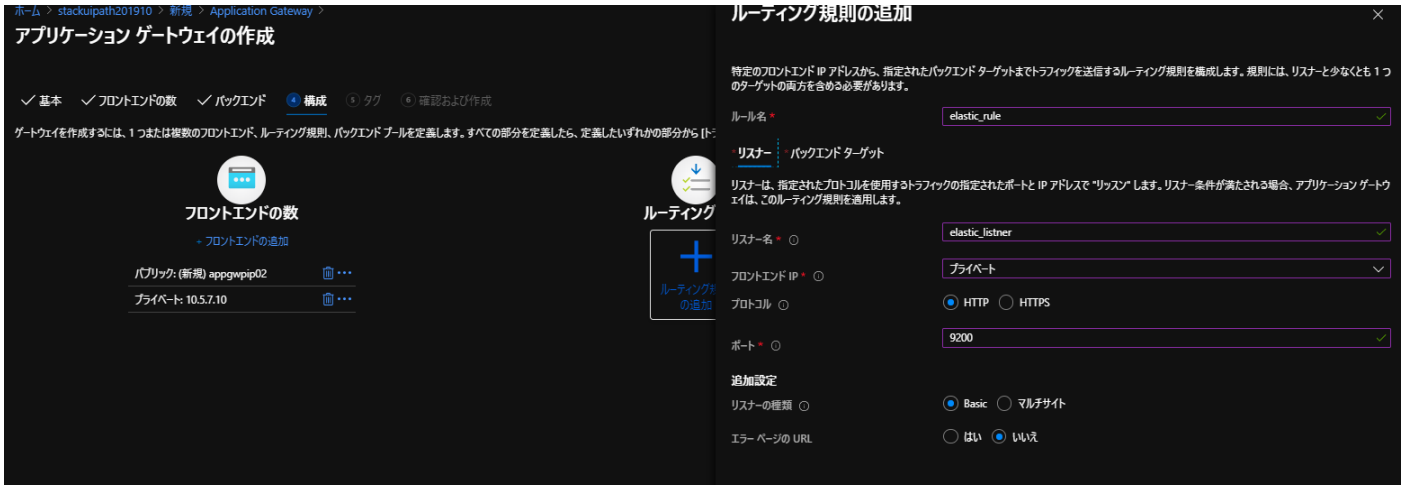
プライベート IP アドレスの指定を Yes にして、任意の Private IP address を入力

Elasticsearch 用の Backend(elastic_backend)を追加

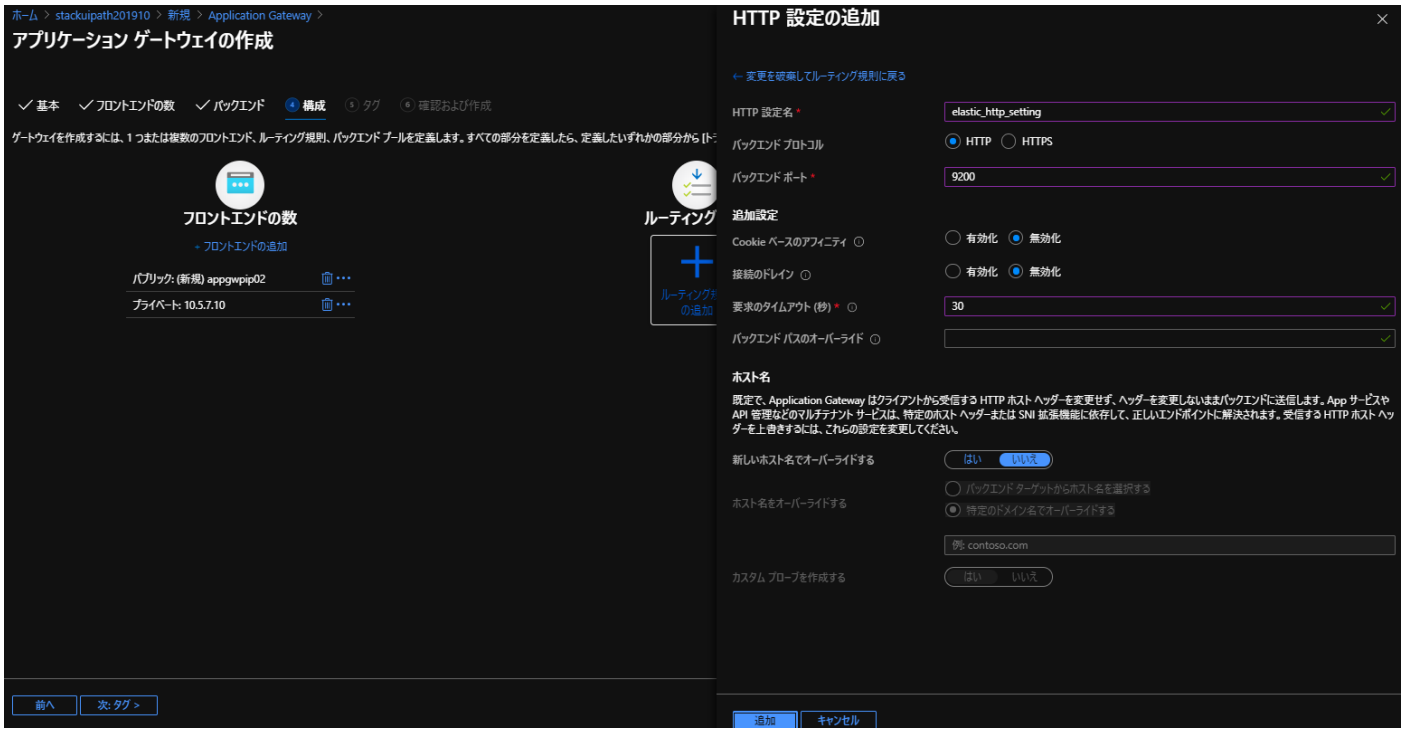
Target Type を「仮想マシン」に設定し、Target で該当 VM がアタッチされている NIC を選択の後、「Add」を押下

Kibana 用の Backend(kibana_backend)も同様にして別 Backend として追加

Routing rules の「Add a routing rule」を押下し、Elasticsearch 用のルーティングルールを追加するために Listener と Backend targets を定義



Backend target を登録するために、HTTP setting 欄「Add new」を選択し、Elasticsearch 用の HTTP Setting を新規作成



同じく Kibana 用の Routing rules を追加

Listener は SSL 終端をさせるために HTTPS を選択し、準備しておいた pfx の証明書を Import。

Elasticsearch と同様に HTTP settings の「Add new」から Kibana 用の HTTP Setting を作成

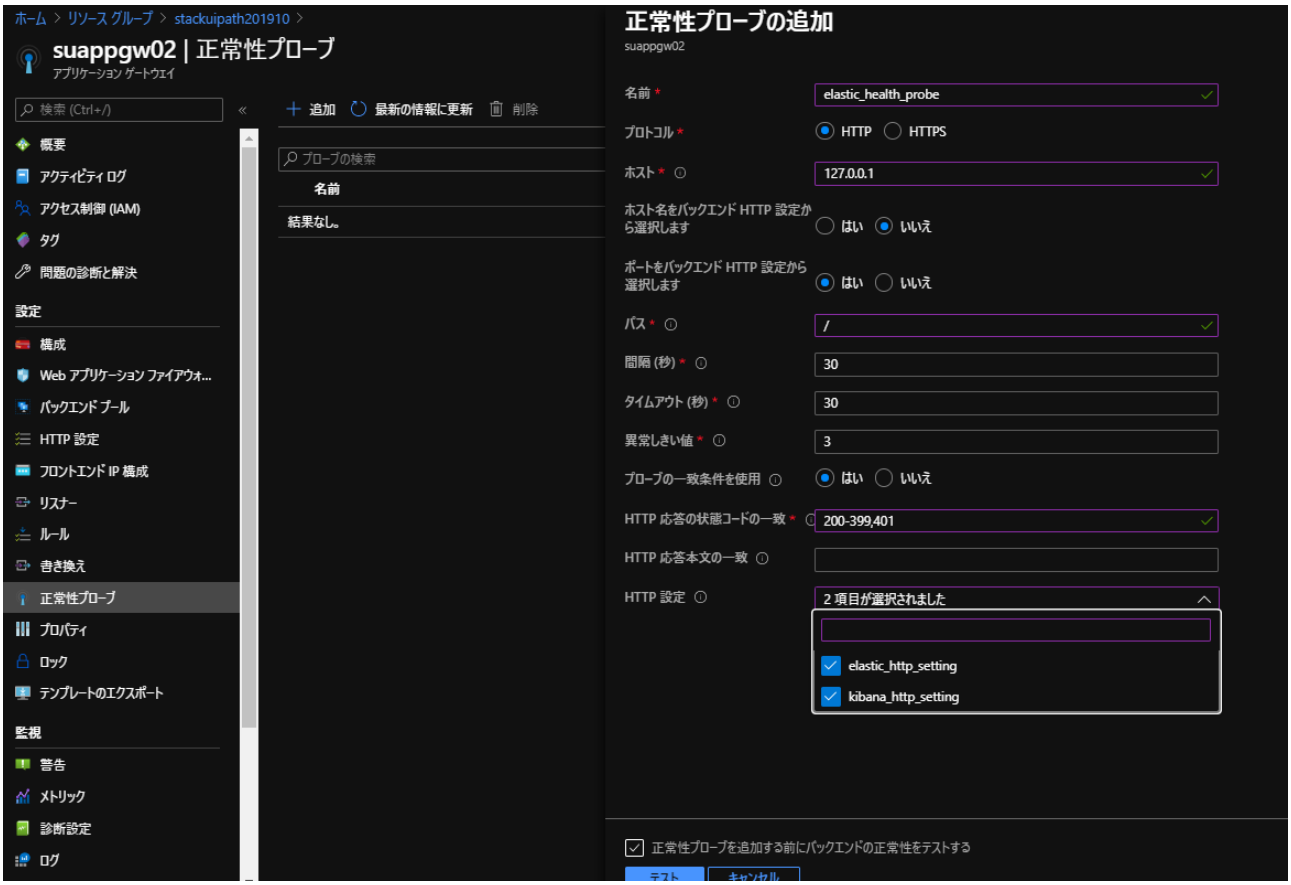
Add を押下し、最終的に以下のような画面となる

最終確認後、Deploy

11.動作確認

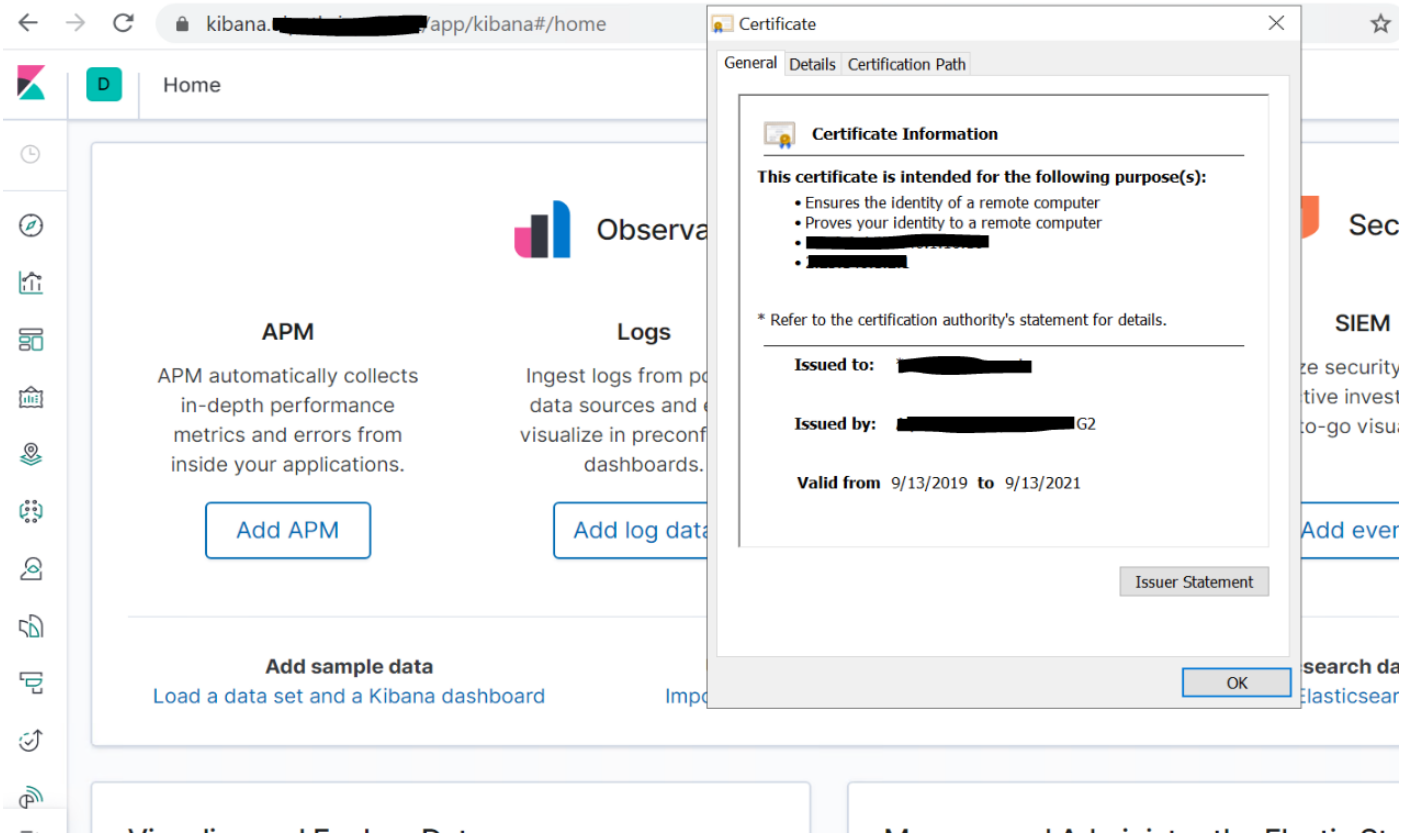
- DNS または hosts に Load Balancer の Private IP を登録後、その登録されている名前で Browser から Elasticsearch に http://<FQDN>:9200 でアクセス
- DNS または hosts に Load Balancer の Private IP を登録後、その登録されている名前で Browser から Kibana に https://<FQDN>でアクセス

(Optional) Elasticsearch Security 機能を有効化した場合の AGW 正常性プローブ設定
正常性プローブ ブレードから「Add」で Custom health probe を以下スクリーンショットのように設定



HTTP response status code match を「200-399,401」とするのは Elastic 社公式ガイドに記載:
A custom health probe is configured that reports healthy for the backend pool for status codes between 200-399, and for status code 401, which may be returned when Elastic Stack Security is enabled, since the health probe makes requests without any form of authentication.
https://www.elastic.co/guide/en/elastic-stack-deploy/7.6/azure-arm-template-load-balancing.html

設定適用後、LB 経由で Kibana にアクセスし、
ログイン認証画面でログインを行い、ログイン後以下の画面が表示されることを確認



12.Orchestrator – Elasticsearch 接続設定

Orchestrator にて収集したロボット実行ログの保存先を SQLServer から Elasticsearch へ変更するため Web.config を以下の通り更新

「9.(Optional) Elasticsearch X-Pack Security 機能有効化」を実施していない場合

Before

```
<target xsi:type="ElasticSearch" name="robotElastic" uri="" requireAuth="false" username="" password="" index="
${event-properties:item=indexName}-${date:format=yyyy.MM}" documentType="logEvent" includeAllProperties="true
" layout="${message}" excludedProperties="agentSessionId,tenantId,indexName" />
～
<logger name="Robot.*" final="true" writeTo="database" />
```

After

```
<target xsi:type="ElasticSearch" name="robotElastic" uri="https://<ApplicationGatewayIP>:<Port>" requireAuth="fal
se" username="" password="" index="${event-properties:item=indexName}-${date:format=yyyy.MM}" documentTyp
e="logEvent" includeAllProperties="true" layout="${message}" excludedProperties="agentSessionId,tenantId,indexNa
me" />
～
<logger name="Robot.*" final="true" writeTo="robotElasticBuffer" />
```

「9.(Optional) Elasticsearch X-Pack Security 機能有効化」を実施した場合

Before

```
<target xsi:type="ElasticSearch" name="robotElastic" uri="" requireAuth="false" username="" password="" index="
${event-properties:item=indexName}-${date:format=yyyy.MM}" documentType="logEvent" includeAllProperties="true
" layout="${message}" excludedProperties="agentSessionId,tenantId,indexName" />
～
<logger name="Robot.*" final="true" writeTo="database" />
```

After

```
<target xsi:type="ElasticSearch" name="robotElastic" uri="https://<ApplicationGatewayIP>:<Port>" requireAuth="tru
e" username="elastic" password="<password of elastic>" index="${event-properties:item=indexName}-${date:form
at=yyyy.MM}" documentType="logEvent" includeAllProperties="true" layout="${message}" excludedProperties="agent
SessionId,tenantId,indexName" />
～
<logger name="Robot.*" final="true" writeTo="robotElasticBuffer" />
```

以上