

[Open in app](#)**Jonathan**[Follow](#)

29 Followers

[About](#)

Azure Kubernetes Service (AKS) on Azure Arc



Jonathan Mar 28 · 3 min read

Azure Arc is one of the latest Azure services that allow administrators to have the capability of managing cloud IaaS, Kubernetes (K8s) clusters and data services with a unified experience, meaning to be able to have an overview from Azure portal. Essentially, every resource needs to install Azure-Arc-related extensions to get connected to the platform. If you would like to learn more about how Azure Arc works, please take a look [here](#).

For this article, we would focus on getting K8s cluster onboarded to Azure Arc. Since I do not have the physical servers to “actually” simulate the K8s-onboarding process, I went through the [Azure Arc Jumpstart tutorial](#) for getting the same onboarding experience.

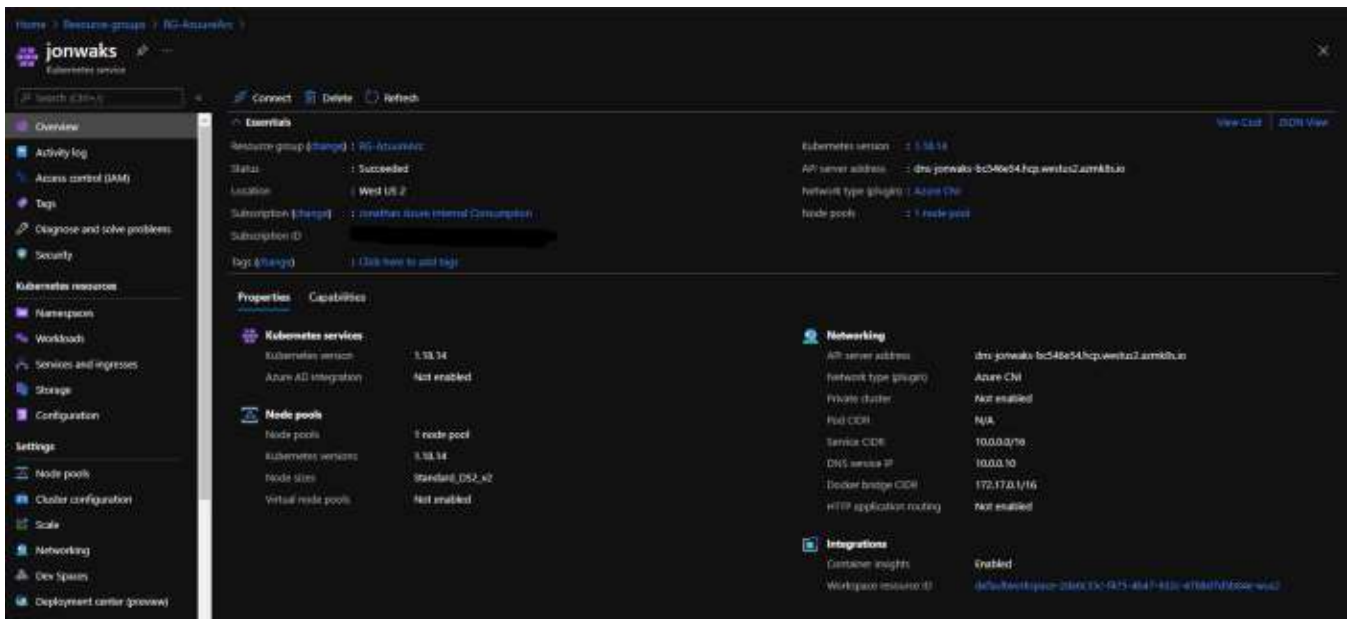
This is meant to be easy as the tutorial already provided the script you would need to execute. This script would not only deploy a whole new AKS cluster in the target resource group but also install all the required extensions within the AKS cluster. So, it is really a one-liner to make life easier.

```
az group create --name Arc-AKS-Demo --location "East US"
az deployment group create \
--resource-group Arc-AKS-Demo \
--name arcaksdemo01 \
--template-uri
```

[Open in app](#)


However, if any of you is like me that always fail to deploy resource through ARM template, worry not. Please follow the steps below to achieve the same goal.

- We first deploy an AKS cluster just like how we would do in normal situation.



- When browsing through '[azuredeploy.json](#)' from above script, you would notice there is a part for executing shell scripts and another part of onboarding the AKS cluster with pre-provisioning Azure Active Directory (AAD) service principal (SP) (yellow part).

```
# Get Environment Variables

USER=$1
SUBSCRIPTIONID=$2
APPID=$3
PASSWORD=$4
TENANTID=$5
RG=$6
LOCATION=$7
VMNAME=$8

# Create .profile script
touch /home/$USER/.bash_profile
chmod +x /home/$USER/.bash_profile

cat <<EOT > /home/$USER/.bash_profile
#!/bin/bash
##Environment Variables
export SUBSCRIPTIONID=$SUBSCRIPTIONID
export APPID=$APPID
export PASSWORD=$PASSWORD
export TENANTID=$TENANTID
```

[Open in app](#)

```
## Configure Ubuntu to allow Azure Arc Connected Machine Agent Installation

echo "Configuring walinux agent"
sudo service walinuxagent stop
sudo waagent -deprovision -force
sudo rm -rf /var/lib/waagent

echo "Configuring Firewall"

sudo ufw --force enable
sudo ufw deny out from any to 169.254.169.254
sudo ufw default allow incoming
sudo apt-get update

echo "Reconfiguring Hostname"

sudo hostname $VMNAME
sudo -E /bin/sh -c 'echo $VMNAME > /etc/hostname'

# Download the installation package
wget https://aka.ms/azcmagent -O ~/install_linux_azcmagent.sh

# Install the hybrid agent
sudo bash ~/install_linux_azcmagent.sh

# Run connect command
sudo azcmagent connect \
  --service-principal-id "${APPID}" \
  --service-principal-secret "${PASSWORD}" \
  --resource-group "${RG}" \
  --tenant-id "${TENANTID}" \
  --location "${LOCATION}" \
  --subscription-id "${SUBSCRIPTIONID}" \
  --tags "Project=jumpstart_azure_arc_servers"
```

** Please ensure all the variables in the shell script has values (pink part). As you are manually executing the script, it would not read the values from the environment, meaning it might require you to manually set the values.

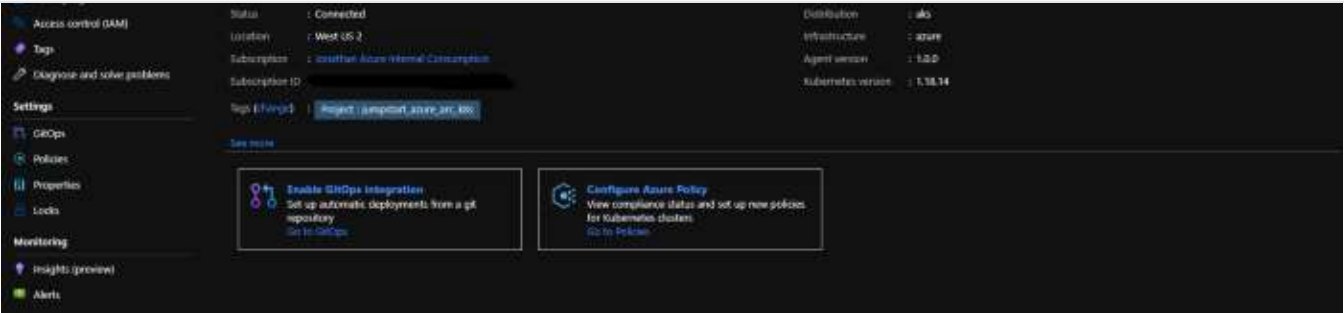
** Please ensure SP has enough permissions to perform actions it needs to perform. To save time, you could temporarily provide contributor role to the entity (blue part).

** The script could also be found in the downloaded Git folder under

```
<whateverdirectory>/azure_arc/azure_arc_servers_jumpstart/azure/linux/arm_template/scripts/install_arc_agent.sh
```

- After the process completes, you would be seeing a new Kubernetes — Azure Arc resource being created in the same resource group. It is leveraging the same AKS icon but with a blue disk underneath, representing this is K8s “on” Azure Arc.

Open in app



Unfortunately, I could not get Linux and Windows servers onboard with ARM template, either and I might create a GitHub issue to address that shortly. Hopefully, by the time you try this, the onboarding experience would feel like a breeze! Happy learning!

[Azure Arc](#) [Azure Kubernetes Service](#) [Onboarding](#) [Arm Templates](#) [Error Handling](#)

[About](#) [Help](#) [Legal](#)

Get the Medium app

