

KNIME Server on Azure Marketplace

KNIME AG, Zurich, Switzerland
Version 4.8 (last updated on 2021-08-04)

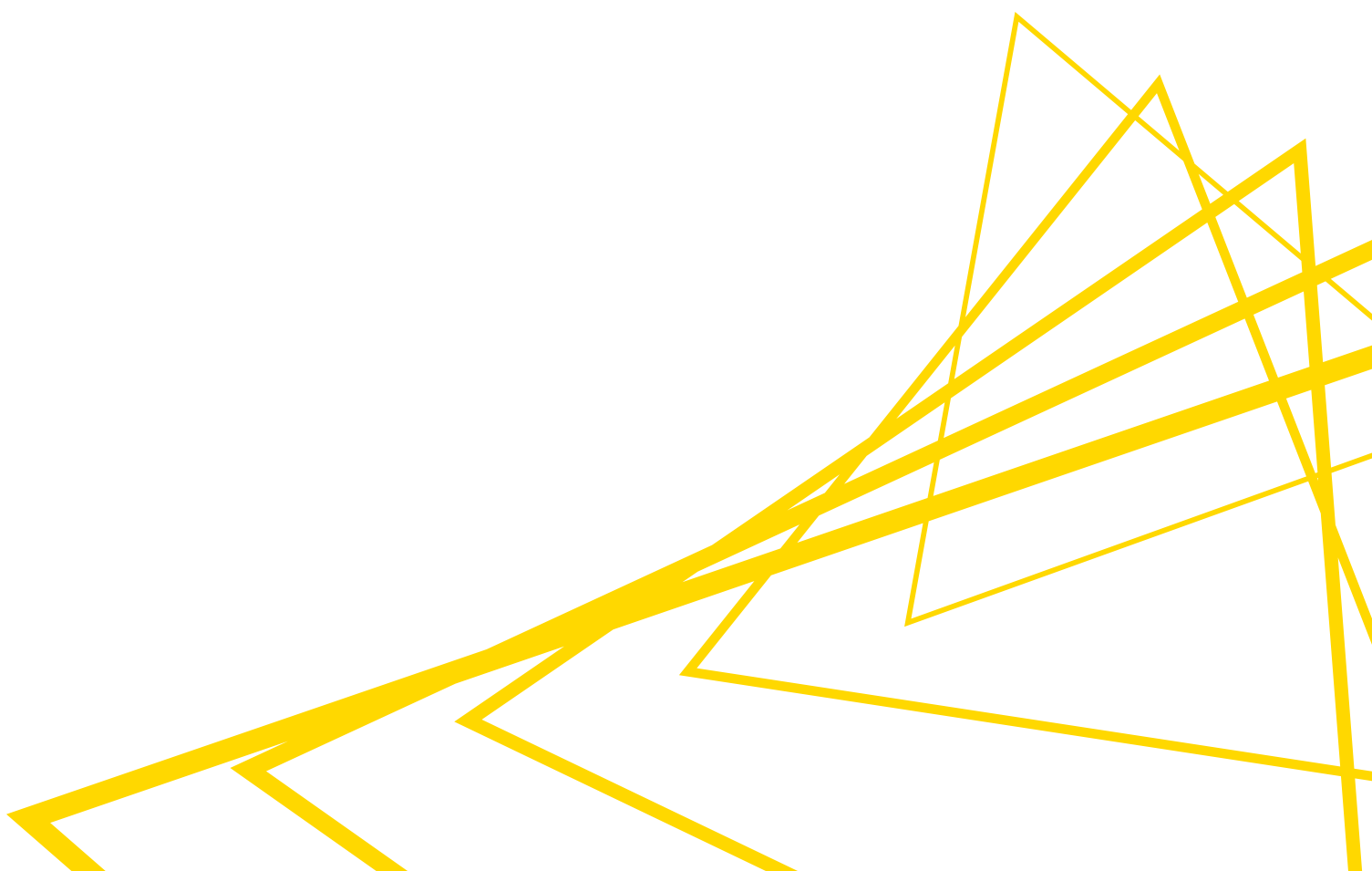


Table of Contents

Introduction.....	1
Deployment on Azure	1
Further reading.....	1
Prerequisites.....	3
Pre-installed software	3
Azure resources	3
Optional external dependencies.....	3
Architecture.....	5
KNIME Server Medium	5
Security	6
Authenticating with Azure	6
Keys and rotation policies	6
Network Security Group access control lists.....	6
Data encryption configuration	6
Audit trail	6
Tagging resources.....	6
Costs	7
Software pricing.....	7
Hardware pricing	7
Required services	7
Estimated costs.....	7
Sizing	9
Azure VM Size selection	9
Disk selection	9
Deployment.....	10
Recommended deployment.....	10
Default KNIME Server password	10
Applying license file to KNIME Server (BYOL)	11
Testing the deployment.....	12
Connecting via the browser	12
Connecting via the Analytics Platform	12
Testing workflow execution.....	12
Azure Resource Manager (ARM) template deployment	13
Creating an ARM template.....	13

Templated VM build	14
Operations.....	16
Availability Zone fault	16
Instance fault.....	16
Application fault	16
Storage capacity	17
Security certificate expirations	18
Backup.....	19
Recovery	19
Routine Maintenance.....	20
Starting KNIME Server	20
Stopping KNIME Server	20
Restarting KNIME Server	20
Bootnote, for versions older than KNIME Server 4.7.....	20
Restarting the executor	20
Key rotation	21
Managing Certificates	21
Default Certificates	21
Apply KNIME Server patches.....	23
Update KNIME Server (feature version)	23
Increasing Azure disk size	23
Emergency Maintenance.....	24
Availability Zone recovery	24
Region recovery.....	24
Support	25
Support costs	26

Introduction

KNIME Server is the enterprise software for team based collaboration, automation, management, and deployment of data science workflows, data, and guided analytics. Non experts are given access to data science via KNIME WebPortal or can use REST APIs to integrate workflows as analytic services to applications and IoT systems. A full overview is available [here](#).

For an overview of use cases, see our [solutions page](#). Presentations at KNIME Summits about usage of the KNIME Server can be found [here](#).

Deployment on Azure

KNIME Server can be launched through the Azure Marketplace. There are several options:

- [KNIME Server Medium](#)
- [KNIME Server \(BYOL\)](#)

For a full list of product offerings including KNIME Analytics Platform, see [here](#).

The KNIME Server (BYOL) instance requires you to Bring Your Own License file to use. To obtain a license file you should contact your KNIME representative, or sales@knime.com.

KNIME Server Medium, and BYOL are single VM instances, and are most easily launched via the [Azure portal](#). If you are familiar with the Azure CLI, or Powershell you may also use this deployment method.

For self-build deployments using a custom base image, you should consult the [KNIME Server Installation Guide](#).

Further reading

If you are looking for detailed explanations around the additional configuration options for KNIME Server, you can check the [KNIME Server Administration Guide](#).

If you are looking to install KNIME Server, you should first consult the [KNIME Server Installation Guide](#).

For guides on connecting to KNIME Server from KNIME Analytics Platform, or using KNIME WebPortal please refer to the following guides:

- [KNIME Explorer User Guide](#)
- [KNIME WebPortal User Guide](#)

There are additional resources such as the [KNIME Server Advanced Setup Guide](#) and [KNIME Server Preview Functionality Guide](#).

Prerequisites

The person responsible for the deployment of KNIME Server should be familiar with basic Azure functionality surrounding configuring Azure VMs. KNIME Server administration requires basic Linux system administration skills, such as editing text files via the CLI, and starting/stopping systemd services.

KNIME Server Medium, and BYOL are single VM images and contain all software requirements.

For self-build instances, please consult the standard [KNIME Server Installation Guide](#).

Pre-installed software

For convenience we have installed and pre-configured:

- OpenJDK 8 (required)
- Anaconda Python
- R
- Chrony
- Azure CLI
- Postfix
- iptables (redirects of requests on port 80, 443 to TomEE running on port 8080, 8443)

Azure resources

Launching a VM requires a new (or available) resource group. Within the resource group a Public IP address, Network Security Group, Virtual Network, Network Interface and Virtual Machine are deployed.

The default Network Security Group will enable HTTP access on port 80, and HTTPS access on port 443. SSH access to administer the server defaults to port 22.

Optional external dependencies

Optionally KNIME Server Large instances (currently available via BYOL license only) may

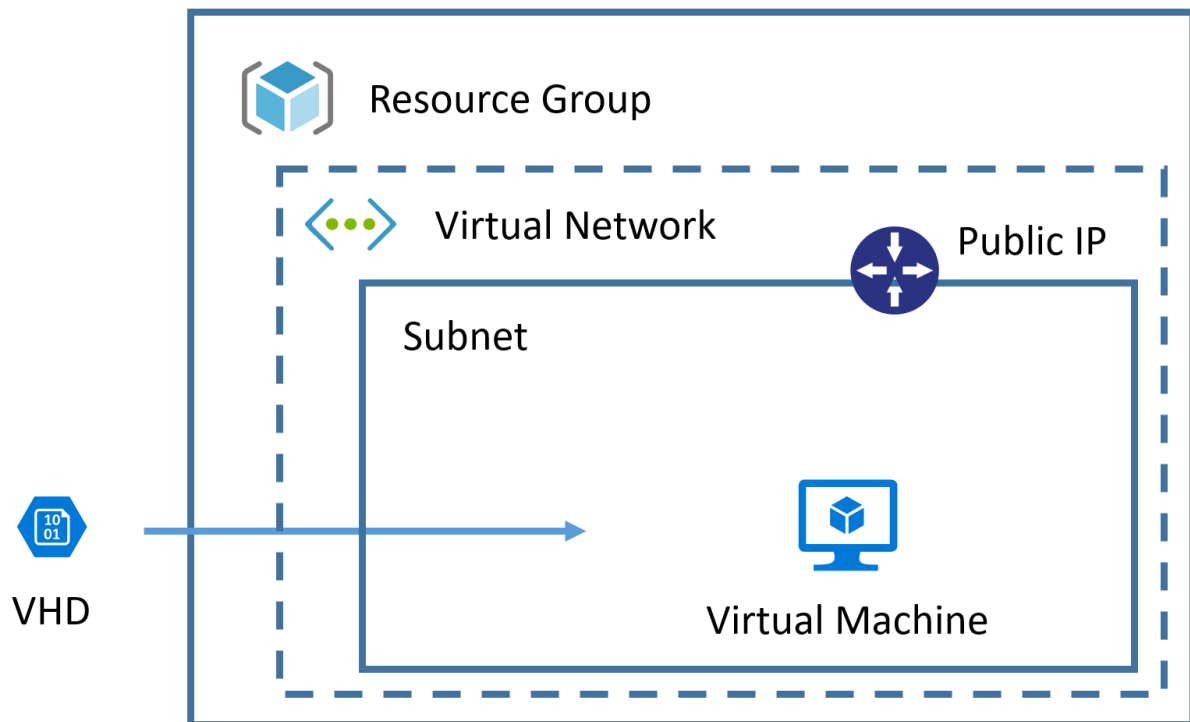
choose to connect KNIME Server to an external LDAP/AD service. Full details are contained in [KNIME Server Advanced Setup Guide](#).

Architecture

An overview of the general KNIME Server architecture is supplied. More detailed description of software architecture can be found in the [KNIME Server Administration Guide](#).

KNIME Server Medium

KNIME Server Medium run as a single VM instance in a single subnet of a Virtual Network. Use of an Public IP is preferred since it simplifies the update/upgrade procedure.



Security

Detailed descriptions of general considerations for KNIME Server security configuration are described in the [KNIME Server Administration Guide](#)

Configurations specific to KNIME Server running on Azure are described below.

Authenticating with Azure

KNIME Server does not require to authenticate with any Azure provided services.

Keys and rotation policies

When SSH access to KNIME Server is required, it is recommended to use an SSH keypair instead of username/password. This can be configured in the Azure portal at launch time.

You are responsible to manage this access key as per the recommendations set within your organisation.

Network Security Group access control lists

The default Network Security Group allows access to the KNIME Server via HTTP, and HTTPS on ports 80 and 443. Additionally advanced admin access via the SSH port 22 is enabled.

Data encryption configuration

Azure managed disks use SSE encryption of disks as default (see [here](#)). You may also wish to enable Azure Disk Encryption (see [here](#)) for all KNIME Server volumes.

Audit trail

KNIME Server log files are accesible via the KNIME Server AdminPortal, or by accessing the files in their standard locations, as described in the [KNIME Server Administration Guide](#)

Tagging resources

You may wish to tag the VM instances and volumes for KNIME Server in order to identify e.g. owner, cost centre, etc. See the [Azure Tagging](#).

Costs

Software pricing

The software pricing for the KNIME Server is defined in the Azure Portal. Questions regarding BYOL licensing should be directed to sales@knime.com.

Hardware pricing

Hardware pricing is defined by Azure. See [Azure Pricing](#)

Required services

- Azure Virtual Machines
- Azure Storage Disk (Managed Disk recommended)
- Data transfer in/out
- Optional: Azure Disk Snapshots

Estimated costs

Costs calculated using the [Azure Cost Calculator](#).

1. Estimated annual costs for a KNIME Server Medium, launched in East US are calculated as follows:

Service	Cost
Software license fees (Hourly fee)	\$ 34803.48
Azure Virtual Machine fee (E8 v3, 1 year reserved)	\$ 3053.04
Azure Storage Disk charges (32Gb + E15 256Gb SSD)	\$ 259.20
Azure Disk snapshots (256Gb/month)	\$ 396.00
Data transfer out (100 Gb/month - Zone 1)	\$ 99.12

Total (annual cost)	\$ 38600.84
---------------------	-------------

Sizing

There is no 'one size fits all' answer to questions around sizing of deployments. The answer will vary depending on your typical workload, number of concurrent users, desired calculation time, and so on. We provide some recommendations to help get started.

KNIME Server Medium are both sold via the Azure Marketplace with built in licenses for 5 named users, and a maximum of 8 cores for workflow execution. Additionally KNIME Server Medium allows 20 consumers to access the KNIME Server WebPortal via the web browser only. Please contact sales@knime.com if you require a larger number of users, consumers, or cores.

Azure VM Size selection

Typically workflow execution speed can benefit from additional available instance RAM. Therefore we recommend the 'Ev3' series, since they provide the best value access to RAM.

The Standard_E8_v3 instance has 64 Gb RAM available, and also 8 CPU cores, thus is the largest instance that KNIME Server Medium can make use of.

Full details of Azure VM sizes can be found [here](#).

Disk selection

The default OS Disk volume is 30Gb Standard SSD or Premium SSD (depending on selected instance type), and in most cases it is not necessary to increase the volume size.

Full details of Azure Disk options are available [here](#).

To help guide the volume size, you will need to consider:

- the size and number of workflows expected to be stored
- the number of jobs executed and job retention duration
- number and size of additional files stored in the workflow repository

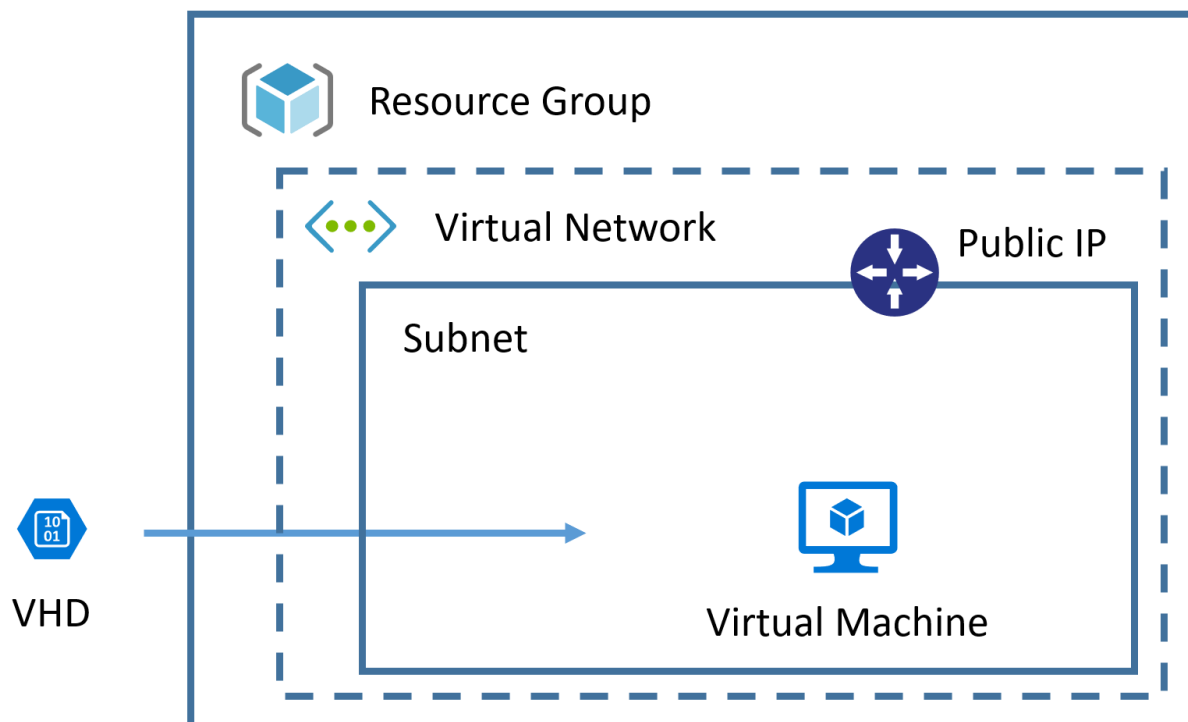
In case you need to add additional storage space later, please see the section [Increasing Azure disk size](#).

Deployment

Deployment of KNIME Server Medium is via a single VM. High-availability options are available as part of KNIME Server Large which is not currently documented for Azure.

Recommended deployment

A typical deployment as per the sizing guidelines in the previous section looks something like:



Default KNIME Server password

We do not set a fixed default password for the KNIME Server since this is a known bad security practice. Therefore the default password is set to the modified MAC address of the VM on the first launch of the KNIME Server. To recreate the password: obtain the MAC address of the VM from the console, strip the colons and convert the text to uppercase.

For example, the MAC address "02:42:ac:11:00:02" needs to be converted to "0242AC110002" to use as the password (minus the quotes).

On first login, you are recommended to create a new admin user, and then remove the `knimeadmin` user.

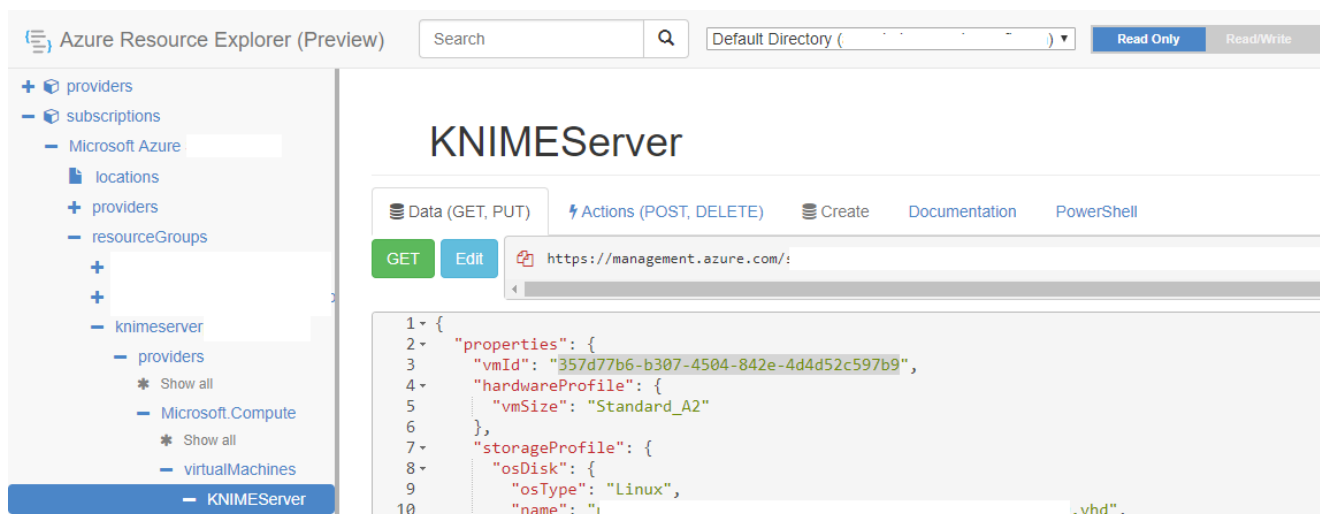
Note that this username/password is for the KNIME Server application, and is different to the KNIME Server VM username/password which can be optionally used for SSH access to the VM.

Applying license file to KNIME Server (BYOL)

For KNIME Server (BYOL) you will need to apply your license file. This can be done by visiting `https://<public-hostname>/knime` from your web browser.

Logging in using the admin username: `knimeadmin`, and password: `<default-password>`, will redirect you to the license upload page. Here you can apply your license file. A valid license file will be immediately applied, and you can begin to use all KNIME Server functionality.

To generate a license file, your KNIME contact will need the `vmId` of the VM. That can be found by browsing to <https://resources.azure.com>



More details on `vmId` can be found [here](#).

You may also choose to login to the instance via SSH and issue the command to get the `vmId`:

```
curl -H Metadata:true "http://169.254.169.254/metadata/instance?api-version=2017-03-01&format=json" | jq -r '.compute.vmId'
```

Testing the deployment

Simple testing of the deployment can be done by logging into KNIME Server WebPortal via the web browser. Certain functionality is only available to test via the KNIME Analytics Platform.

Connecting via the browser

Once you have launched the KNIME Server VM, the resulting instance will automatically start the KNIME Server. The KNIME Server WebPortal is available in the browser at `https://<public-hostname>/knime`

Connecting via the Analytics Platform

Access to the KNIME Server from the KNIME Analytics Platform is via the KNIME Explorer. Full documentation is available in the [KNIME Explorer User Guide](#) Use the mountpoint address: `https://<public-hostname>/knime` Username is `knimeadmin`, and password is `<default-password>` unless you already changed them.

Testing workflow execution

Click on any workflow from the WebPortal repository tree, and wait for the page to load. If the 'Start' button appears then workflow execution is working as expected. For automated testing strategies, see the section: [Automated testing](#).

Azure Resource Manager (ARM) template deployment

Whilst it is easy and convenient to deploy the KNIME Server through the Azure Portal, there are strong arguments for using a templated deployment using **ARM templates**.

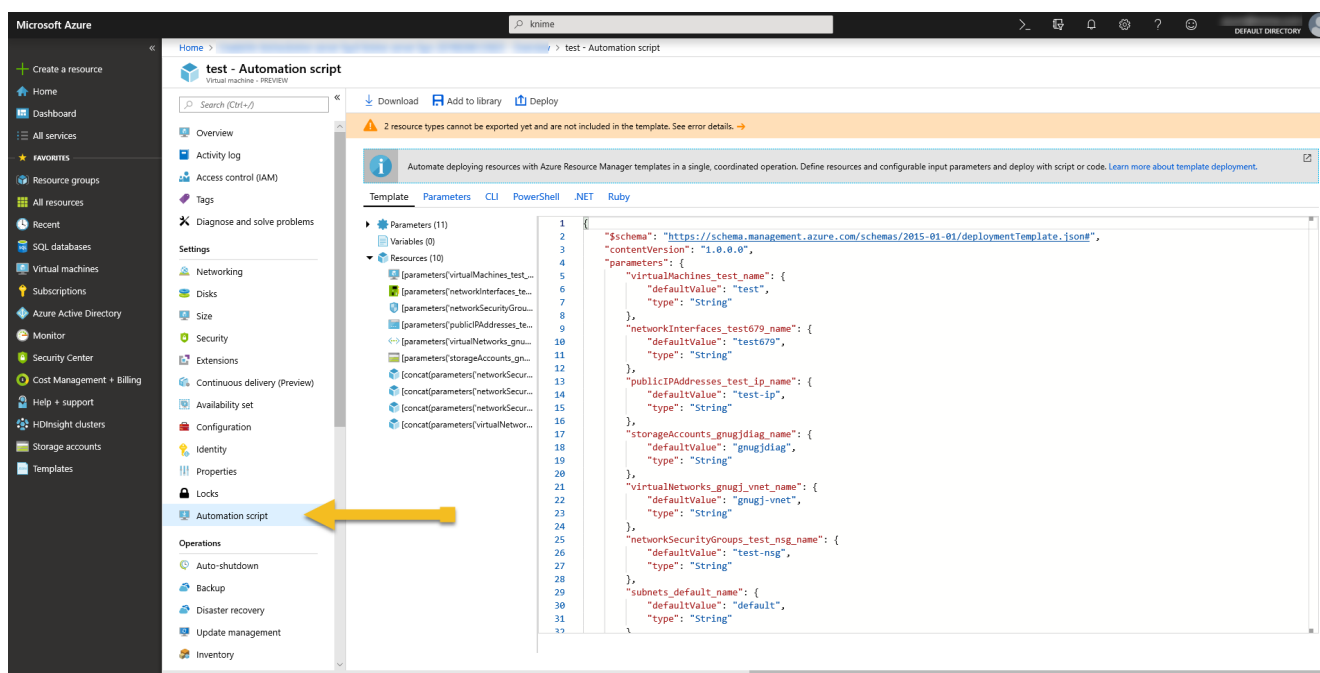
Azure Resource Manager (ARM) templates allow to describe the full state of the final deployment such that it may be redeployed identically in the future. Similarly it is possible to alter a master template which allows for easy reuse of shared configurations, and customizations where required.

For an overview of the types of deployments possible with an ARM template see the Azure documentation **Quickstart templates**.

Creating an ARM template

The ARM template describes the Azure infrastructure that is required to be launched for the KNIME Server.

If you already have an existing deployment that you deployed through the Azure Portal, it is possible to export the ARM template from the Azure Portal by navigating to the Virtual Machine, and selecting **Automation Template** as shown in the below image.



Below is an example template that could be used as the starting point for a simple deployment.

arm_template_example

Example ARM template

The template itself is parametrized by the following file.

arm_template_parameters_example

Example ARM template parameters

Splitting the template from the parameters allows to describe multiple deployments, e.g. dev and prod using a single template, with a parameters file for each. That can help to make sure that the setup in the development environment is the same as the production environment.

Templated VM build

In addition to the Azure Marketplace images for KNIME Server (Medium, or BYOL) you have the option to install and configure KNIME Server from scratch. In this case the installation process is described in the [KNIME Server Installation Guide](#). Additional configurations of KNIME Server are described in the [KNIME Server Administration Guide](#).

In the case where you want to automate the build of a KNIME Server 'Golden Image' you should consider the above two documents as the required information. It will then be necessary to adapt your internal build process to follow this procedure. None of the tools mentioned are required, and you may choose to use alternatives.

We describe in brief detail the steps that we follow at KNIME to build the Azure Marketplace images. We use the tool [Packer](#) to automate the process.

The description is not intended to be an exhaustive list, but an overview of the kinds of things that you will need to consider.

We follow steps such as:

- Define 'base' image. We choose Ubuntu 16.04 LTS.
- Apply latest OS patches
- Upload configuration files (preferences.epf, knime.ini, license.xml, autoinstall.xml, etc)
- Create VM user (knime)
- Install required dependency (Java JDK 8)
- Run automated KNIME Server installer
- Install optional dependencies (Python, R, Chrony, etc)
- Configure Port Forwarding/Firewall or Front-end webserver

- Cleanup image and generalize

Operations

As part of any KNIME Server deployment you should consider monitoring your service for availability. KNIME Server has several endpoints that can be used to determine the system health.

Availability Zone fault

Since KNIME Server Medium runs in a single Availability Zone an Availability Zone fault will be detected by the application fault detection method described below.

Instance fault

A VM fault can be detected using the standard Azure techniques.

Application fault

A simple REST call to the deployed KNIME Server should always return a 200 response with a payload similar to:

```
curl https://<public-hostname>/knime/rest
```

rest_response

```
{
  "@controls" : {
    "self" : {
      "href" : "https://<public-hostname>/knime/rest/",
      "method" : "GET"
    },
    "knime:v4" : {
      "href" : "https://<public-hostname>/knime/rest/v4",
      "title" : "KNIME Server API v4",
      "method" : "GET"
    }
  },
  "version" : {
    "major" : 4,
    "minor" : 8,
    "revision" : 0,
    "qualifier" : ""
  },
  "mountId" : "<public-hostname>",
  "@namespaces" : {
    "knime" : {
      "name" : "http://www.knime.com/server/rels#"
    }
  }
}
```

A different response indicates a configuration issue, or application fault.

It is also possible to test for executor availability. This requires authenticating against the KNIME Server and calling the following REST endpoint.

```
curl -X GET "https://<public-hostname>/knime/rest/v4/repository/Examples/Test Workflows
(add your own for databases)/01 - Test Basic Workflow - Data
Blending:execution?reset=true&timeout=300000" -H "accept:application/vnd.mason+json"
```

Storage capacity

You may monitor the storage capacity of the Azure disk volumes (OS Disk and Data disk) using standard techniques and services such as Azure Monitor. For more details see [here](#).

We recommend triggering an alarm at <5% free space on either volume.

Security certificate expirations

Certificate expiration will be caught if the basic server check fails with an HTTP 400 status code.

Backup

KNIME Server can be backed up subject to the information available in the [KNIME Server Administration Guide](#).

It is recommended to make use of the Azure Snapshot functionality. See the Azure documentation section on [taking Azure disk snapshots](#).

Recovery

To restore an Azure Disk Snapshot, see the Azure documentation section on [restoring Azure disk snapshots](#).

Routine Maintenance

Starting KNIME Server

KNIME Server starts automatically when the instance starts using standard systemd commands. Once the TomEE application has started successfully, it will automatically launch and executor. This means that in normal operation you will not need the below command.

In the case that you need to start a stopped KNIME Server, it may be started using the following command at the terminal:

```
sudo systemctl start knime-server.service
```

Stopping KNIME Server

Stop the KNIME Server by executing the command:

```
sudo systemctl stop knime-server.service
```

Restarting KNIME Server

Restart the KNIME Server by executing the command:

```
sudo systemctl restart knime-server.service
```

Bootnote, for versions older than KNIME Server 4.7

Note that starting, stopping and restarting differs from version 4.7 and older of KNIME Server, where knime-server.service was replaced with apache-tomee.service

Restarting the executor

It is possible to restart the executor by issuing the following command:

```
sudo -u knime touch /srv/knime_server/rmirestart
```

This will launch a new executor, leaving the existing executor running. All existing jobs will continue to run on the old executor, and all new jobs will be launched on the new executor. That is helpful when updating executor preference files without needing to interrupt existing running jobs. When the `rmirestart` file is automatically deleted, the new executor has been launched.

It is possible to perform a hard kill on a running instance, by issuing the command:

```
sudo -u knime kill -9 <PID>
```

where `<PID>` is the process ID of the running executor. You can find the `<PID>` by running:

```
ps aux | grep knime
```

and looking for the process(es) that are not the `apache-tomee` instance.

Key rotation

Managing SSH keys for accessing the KNIME Server is detailed [here](#).

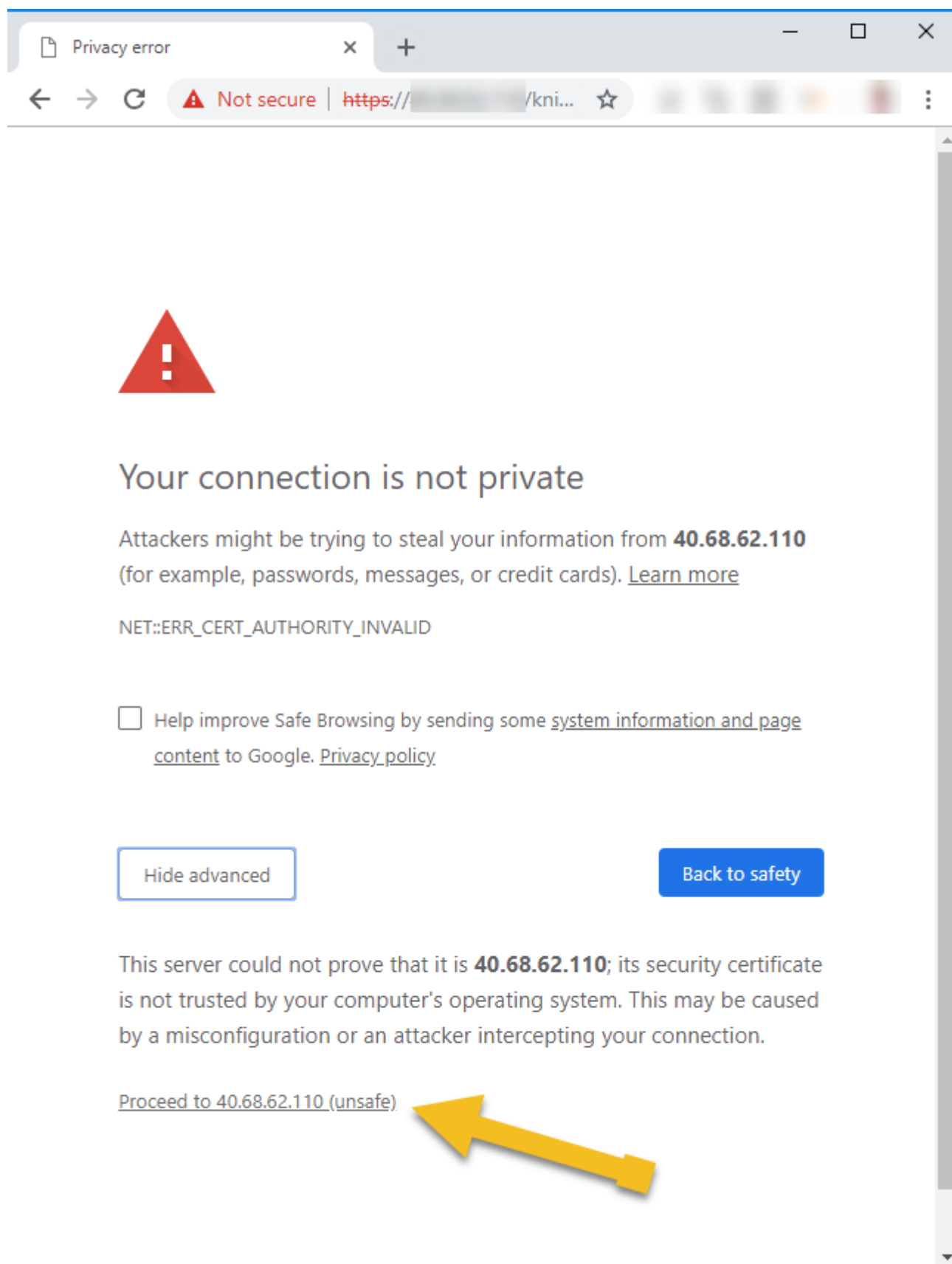
Managing Certificates

Detailed steps for managing the SSL certificate for KNIME Server can be found in the [KNIME Server Administration Guide](#)

Default Certificates

KNIME Server ships with a default SSL certificate. This allows for encrypted communication between client and server. However, since the certificate cannot be generated in advance for the server that you are running on, it will not be recognised as a valid certificate. Therefore, we recommend managing your own certificate as per the guidelines in the [Managing Certificates section](#).

When testing with the default certificate, modern browsers will issue a warning as below. Choosing to ignore the warning, will allow you to access the KNIME WebPortal for testing.



Apply KNIME Server patches

Patches to the KNIME Server are announced on the KNIME Server Forum. You may subscribe to the [topic](#). Details of update procedure are described in the [KNIME Server Update Guide](#).

Update KNIME Server (feature version)

To make a feature update you have the option to follow the instructions in the [KNIME Server Update Guide](#).

Increasing Azure disk size

It is possible to increase the size of the workflow repository disk size (default size: 30 Gb) after an instance has been launched. Follow the instructions [here](#).

Emergency Maintenance

In case KNIME Server is not available due to degraded performance of an Availability Zone (AZ), VM fault, etc. It is possible to restore a snapshot and launch a new instance.

Availability Zone recovery

Availability Zone recovery is managed by launching a new instance into an unaffected Availability Zone, using a **recent snapshot**.

Then attach the elastic IP from the affected instance to the new instance.

Region recovery

Region recovery is managed by launching a new instance into an unaffected region, using a **recent snapshot**.

Then attach the elastic IP from the affected instance to the new instance.

Support

KNIME Server Medium support is supplied via contacting the support@knime.com email address.

When contacting KNIME Support you will need to include your Product Code, Instance ID, and Azure Account ID.

We aim to respond to your question in under 48 hours.

Support costs

If you require additional support, please contact sales@knime.com for further information.

KNIME AG
Technoparkstrasse 1
8005 Zurich, Switzerland
www.knime.com
info@knime.com