SAS® Viya® 4: Deployment Guide

SAS® Documentation
March 16, 2020

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2020. *SAS® Viya® 4: Deployment Guide*. Cary, NC: SAS Institute Inc.

SAS® Viya® 4: Deployment Guide

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

May 2020

4.0.1-P1:dplyml0phy0dkr

Contents

Early Adopter Software v

Chapter 1 / Introduction **1**
How Deployment Works 1

Chapter 2 / System Requirements **3**
Virtual Infrastructure Requirements 3
Hardware Requirements 5
Data Source Requirements 6
Identity Provider Requirements 6
SAS Infrastructure Data Server Requirements 7
Client Requirements 8

Chapter 3 / Pre-installation Tasks **9**
Retrieve Required Files 9

Chapter 4 / Installation **11**
Create the kustomization.yaml File 11
Deploy the Software 18
Sign In as the sasboot User 19
Configure LDAP 20

Chapter 5 / Validating the Deployment **23**
Verify SSSD 23
Verify SAS/ACCESS Interface to Oracle 23
Verify PC Files Connectivity 24

Chapter 6 / Completing the Deployment **25**
Administration Documentation 25

Chapter 7 / Managing Your Software **27**
Latest Requirements 27
Applying a New License for Your SAS Viya Software 27
Updating Your SAS Viya Software 29

Appendix 1 / Using a Mirror Repository **31**
Create a Mirror Repository 31
Add a Mirror Repository to kustomization.yaml 31

Early Adopter Software

THIS DOCUMENTATION FOR AN EARLY ADOPTER PRODUCT IS A PRELIMINARY DRAFT AND IS PROVIDED BY SAS INSTITUTE INC. ("SAS") ON AN "AS IS" BASIS WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTIBILITY AND/OR FITNESS FOR A PARTICULAR PURPOSE. SAS does not warrant that this documentation is complete, accurate, or similar to that which may be released to the general public, or that any such documentation will be released. The company shall not be liable whatsoever for any damages arising out of the use of this documentation, including any direct, indirect, or consequential damages. SAS reserves the right to alter or abandon use of this documentation at any time.

1

Introduction

How Deployment Works 1
The Basics 1
Deployment Options to Consider 1

How Deployment Works

The Basics

SAS Viya applications are deployed as containerized applications to a Kubernetes cluster. Here a few items to consider:

- The deployment assets provide access to the images in the SAS image repository. Using the deployment assets and Kustomize, you will provide site-specific information, generate a Kubernetes manifest, and run the kubectl apply command to create your SAS Viya deployment using the manifest.
- To generate the manifest, you must create a kustomization.yaml file that is custom to your environment. This file includes declarations for the required bases and overlays, ingress controller or service mesh options, and more. Instructions for what must be included in the file are included in this guide.

Note: The kustomization.yaml file will be reused for future updates to your SAS Viya deployment.

Deployment Options to Consider

Deployment options are configured as overlays in the kustomization.yaml file.

Option	Explained
SMP or MPP CAS server	In the SAS Viya platform, the SAS Cloud Analytic Services (CAS) Server provides the run-time environment where data management and analytics take place. The CAS server can be deployed to a single node or in a distributed fashion across multiple nodes. Deploying the CAS server on a single node allows for symmetric multi-processing (SMP) by users. A single-node CAS server performs serial loads of data into memory from a supported data source. The in-memory analytic features of a distributed CAS server are available to the single-node CAS server. Distributing the CAS server across multiple nodes allows for massively parallel processing (MPP) by users. An advantage to MPP is that, whenever possible, data is loaded into memory in parallel, which can result in faster load times.

2

System Requirements

Virtual Infrastructure Requirements 3
Overview of Infrastructure Requirements 3
Host Machine Requirements 4

Hardware Requirements 5
Hardware Requirements 5
Storage Requirements 5

Data Source Requirements 6

Identity Provider Requirements 6

SAS Infrastructure Data Server Requirements 7
Deployment Options 7

Client Requirements 8
Web Browsers 8
Mobile Platform and Touchscreen Support 8
Screen Resolution 8

Virtual Infrastructure Requirements

Overview of Infrastructure Requirements

Requirements for the deployment of SAS Viya 4.0.1 are minimal because most of the required components are included in the Docker containers that make up your SAS Viya deployment.

The topics in this section will assist you in preparing your environment for the deployment.

Host Machine Requirements

The kubectl Machine

The deployment requires a machine from which the Kubernetes command-line interface, kubectl, manages a Kubernetes cluster. This machine can be running Linux, Windows, or Macintosh.

Kubectl 1.14 or later must be installed on this machine.

After kubectl is installed, note the location of the Kubernetes configuration file for use during the deployment process.

In addition, the kubectl machine requires kustomize 3.4.0 or later. Kustomize is a Kubernetes enhancement. You can find it on GitHub.

The Kubernetes Cluster

The Kubernetes cluster is where your SAS Viya software runs. Kubernetes 1.13 or later is supported.

To determine your version of Kubernetes, run the following command:

```
kubectl version
```

Microsoft Windows operating systems are not currently supported.

Cluster Requirements

Your SAS Viya deployment will require a namespace in the cluster. You can check to see if your cluster has a namespace by running the following command:

```
kubectl get namespaces
```

The result is a list of namespaces like the following. Verify that the namespace you want to use is included in the list.

```
NAME              STATUS   AGE
default           Active   11h
kube-node-lease   Active   11h
kube-public       Active   11h
kube-system       Active   11h
viya              Active   20h
```

An Ingress controller, such as NGINX or Istio, that is supported by Kubernetes is required. If Istio is the Ingress controller that is used, it must be version 1.2.x or later.

Hardware Requirements

Hardware Requirements

The following table lists products that can be separately licensed and indicates the minimum RAM and number of CPU cores to support individual components. The combined capacity of the Kubernetes nodes should add up to the numbers in the table. Testing was performed with a full deployment. These guidelines do not apply to a programming-only deployment.

The table represents what is required to start all system services and to enable a single user to operate against a small sample data set in order to validate operational functionality. These out-of-the-box requirements should be increased for larger deployments.

Additional RAM should be added based on the expected amount of data that will be processed. More resources are required for multiple-user, production-scale deployments that use large data sets. These guidelines do not attempt to account for all ordering scenarios, but instead are intended to illustrate typical software orders.

Table 2.1 *Minimum Hardware Requirements for Each Product*

Products	RAM (GB)	CPU Cores
SAS Visual Analytics	40	8
SAS Visual Analytics and SAS Visual Statistics	40	8
SAS Visual Analytics, SAS Visual Statistics, and SAS Visual Data Mining and Machine Learning	56	12
SAS Visual Analytics and SAS Data Preparation	40	8

Storage Requirements

The minimum amount of disk space that is required for the installation and for logging is 48 GB. Therefore, the minimum combined capacity of the worker nodes should be 48 GB.

Persistent storage is required. Ensure that you have a storage class defined and set as your default. Confirm you have a default storage class defined with the following command:

```
kubectl get storageclass
```

Data Source Requirements

SAS Viya supports only Oracle as an external data source.

If you use Oracle as an external data source, SAS Viya requires the following Oracle components:

- Oracle Database 11gR2 or later
- Oracle Client 11gR2 or later (64-bit libraries)

SAS/ACCESS Interface to Oracle supports the following cloud variants of Oracle:

- Amazon RDS Oracle 11gR2 or later
- Oracle Cloud Platform 11gR2 or later

Obtain the path to the volume on the Oracle server to which you want to point your SAS Viya deployment. This information is used later in the deployment.

Identity Provider Requirements

SAS Viya supports LDAP for user and group identities and authentication. The following requirements apply to your LDAP server:

- SAS Viya must have Read access to your LDAP server.
- SAS Viya requires a userDN and password in order to bind to the LDAP server. Anonymous binding is supported for clients that are authenticating to the LDAP server.
- If the mail attribute is specified for LDAP accounts, it must have a non-null value that is unique for each user.
- LDAPS is supported, but the required certificates are not configured automatically by the deployment process.

You can set up your LDAP users with SAS Environment Manager after your software has been deployed or use the sitedefault.yml file before the deployment.

SAS Infrastructure Data Server Requirements

Deployment Options

SAS Infrastructure Data Server stores SAS Viya user content, such as reports, authorization rules, selected source definitions, attachments, audit records, and user preferences. SAS Viya uses High Availability (HA) PostgreSQL for SAS Infrastructure Data Server. Take a few minutes to consider your options for deploying a PostgreSQL instance to support the data server.

Internal Versus External PostgreSQL Instances

An instance of PostgreSQL is required for the SAS Data Infrastructure Server. You can let SAS automatically deploy an instance of PostgreSQL (described as an *internal* instance), or you can provide your own PostgreSQL (an *external* instance). Before deploying your software, you must decide which type of instance you want to have.

> **IMPORTANT** After your SAS Viya software has been deployed, you cannot change the PostgreSQL deployment type without redeploying the entire software package.

Internal PostgreSQL

If you decide to deploy an internal instance of PostgreSQL for SAS Infrastructure Data Server, SAS configures and maintains the deployment for you.

SAS deploys a comprehensive PostgreSQL container operator suite provided by Crunchy Data for this purpose.

Requirements for External PostgreSQL

If you decide to deploy an external instance of PostgreSQL, you are responsible for configuring and maintaining the PostgreSQL deployment.

PostgreSQL version 12 is the only supported version of PostgreSQL for SAS Viya.

The database user must have the following permissions:

- SUPERUSER
- INHERIT
- CREATEDB
- CREATEROLE

- REPLICATION (required if the PostgreSQL environment is set up with replication enabled)

The external PostgreSQL instance must be up and running with the database user created before SAS Viya is deployed.

Client Requirements

Web Browsers

End users can access the product user interfaces for SAS Viya applications from a desktop computer, using a supported web browser. Because SAS software is not installed on this machine, the requirements are minimal. UNIX and 64-bit Windows operating systems are supported.

Some SAS Viya user interfaces include some advanced features that require recent versions of popular web browsers. For information about supported web browsers and the corresponding platforms to access SAS user interfaces, see: https://support.sas.com/en/documentation/third-party-software-reference/viya/35/support-for-web-browsers.html.

Mobile Platform and Touchscreen Support

The SAS Visual Analytics Apps run natively on iOS, Android, and Windows 10, and provide the ability to view and explore reports using a touchscreen.

Some SAS Viya user interfaces are not currently supported on mobile devices.

For more information about mobile device support, see: https://support.sas.com/en/documentation/third-party-software-reference/viya/35/support-for-web-browsers.html.

Screen Resolution

The minimum screen resolution for each client machine that will access the SAS Viya user interfaces is 1280 x 1024.

3

Pre-installation Tasks

Retrieve Required Files 9

Retrieve Required Files

The required deployment assets are delivered from the My SAS site in a .tgz file.

1 Create a directory at the root level of a machine that can be reached by your kubectl machine or on the kubectl machine itself:

```
mkdir directory-name
```

SAS recommends that you name the directory `deploy`, but you should use a name that is meaningful to you. The directory will be referred to as $deploy throughout the rest of this deployment guide. Replace the string with the directory name you choose.

2 Click the **Get Started** link provided in your SOE.

3 On the My SAS web page that opens, expand the information for the order listed in your SOE by clicking the plus sign (“+”).

4 Under **Order Assets**, click **Download Deployment Assets**.

5 Save the .tgz file from the My SAS page to the directory you created in step 1.

6 Extract the files from the .tgz in the same directory:

```
tar xvfz file-name.tgz
```

7 The result is a directory structure that looks like this:

```
$deploy
   bundles
      default
      programming
      data-agent-on-premises
```

The subdirectories represent the deployment types that are available for the software in your order. Not all of the deployment types may be present if your order does not support one or more of them. The deployment types are described below:

Table 3.1 *Deployment Types*

Deployment Type	Description
default	Includes all the software to which you are entitled.
programming	Excludes SAS Drive, most graphical user interfaces, and most services. It is the simplest and smallest type of deployment.
data-agent-on-premises	Includes only SAS Data Agent, to be deployed on the opposite side of a firewall from SAS Data Preparation (a *remote* deployment)

For more information about how to use the files in the subdirectories, see "Create the Initial kustomization.yaml File" on page 11.

4

Installation

Create the kustomization.yaml File 11
Create the Initial kustomization.yaml File 11
Add a sitedefault.yml File to Your Deployment 12
Specify SMP or MPP CAS 13
Configure PostgreSQL 13
Configure CAS Settings 13
Create the Kubernetes Manifests 18
Deploy the Software 18
Sign In as the sasboot User 19
Configure LDAP 20
Configure the Connection to Your Identity Provider 20
Set Up Administrative Users 22
Sign In Using LDAP Credentials 22

Create the kustomization.yaml File

The kustomization.yaml file is the means by which you customize your Kubernetes deployment and allocate resources. For more information about how the kustomization.yaml file is used, see "How Deployment Works" on page 1.

Create the Initial kustomization.yaml File

1 Change to the directory where you saved the deployment assets in "Retrieve Required Files" on page 9:

```
cd /$deploy
```

2 Create a kustomization.yaml file. Copy the following as the basis of the kustomization.yaml file:

```
namespace: name-of-namespace
resources:
- bundles/deployment-type/bases/sas
# Use this line only if you are using Istio as a service mesh
# - bundles/deployment-type/overlays/network/istio
# Remove the following line if you are using Istio as a service mesh
- bundles/deployment-type/overlays/network/ingress
- bundles/deployment-type/overlays/cas-smp
- bundles/deployment-type/overlays/internal-postgres
- bundles/deployment-type/overlays/crunchydata
transformers:
- bundles/deployment-type/overlays/required/transformers.yaml
- bundles/deployment-type/overlays/internal-postgres/internal-postgres-transformer.yaml
# Mount information
# - mount-file-name.yaml
# License information
# secretGenerator:
# - name: sas-license
#   type: sas.com/license
#   behavior: merge
#   files:
#   - SAS_LICENSE=license.jwt
configMapGenerator:
- name: ingress-input
  behavior: merge
  literals:
  - INGRESS_HOST=name-of-ingress-host
- name: sas-shared-config
  behavior: merge
  literals:
  - SAS_URL_SERVICE_TEMPLATE=http://name-of-ingress-host:port
```

Replace *deployment-type* with the subdirectory that represents the deployment type you are using. For more information about the deployment type, see "Retrieve Required Files" on page 9.

Add a sitedefault.yml File to Your Deployment

If you want to load a SAS Viya 3.X sitedefault.yml file in your deployment, add the following content to the kustomization.yaml file in the configMapGenerator section:

```
- name: sas-consul-config
  behavior: merge
  files:
    - SITEDEFAULT_CONF=sitedefault.yml
```

Ensure the sitedefault.yml file is in the same directory as the kustomization.yaml file.

Note: The sitedefault.yml sample file is not currently available in SAS Viya 4.0. When it is available, the sample can be used to create a sitedefault.yml to bulk load defaults for the Viya environment. After the initial deployment, you cannot simply modify sitedefault.yml to change an existing value and deploy the software again. You can modify sitedefault.yml only to set property values that have not already been set. Therefore, SAS recommends that you do not use sitedefault.yml, except where specifically described in this document.

Specify SMP or MPP CAS

Your deployment of Cloud Analytic Services (CAS) can be performed on a single node (SMP) or across several machines (MPP). Go to the **bundles/*deployment-type*/overlays/cas-*smp-or-mpp*** directory and follow the directions in the README file located there to modify the appropriate files.

Replace *smp-or-mpp* with `smp` if you are deploying a single CAS node or use `mpp` if you are deploying a distributed CAS. For more information about SMP and MPP deployments of CAS, see "Deployment Options to Consider" on page 1.

Note: If you use the kustomization.yaml from Step 2 on page 11, it includes the content required for deploying SMP CAS.

Configure PostgreSQL

Based on your decision about your PostgreSQL instance (see " Deployment Options" on page 7), you must perform steps to deploy PostgreSQL (internal) or connect to an existing PostgreSQL instance (external). Go to the **bundles/*deployment-type*/overlays/*internal-or-external*–postgres** directory and follow the directions in the README file located there. Your decision about an internal or external instance determines which directory you should use.

Note: If you use the kustomization.yaml from Step 2 on page 11, it includes the content required for deploying internal PostgreSQL.

Configure CAS Settings

Mount persistentVolumeClaims and Data Connectors for the CAS Server

1 Create a new .yaml file with a name that reflects the action you want to perform. A good example is cas-mount.yaml, but you should use a name that is meaningful to you.

2 In the new .yaml file, add the following content:

```
# mount-file-name.yaml
# Add additional mount
apiVersion: builtin
kind: PatchTransformer
metadata:
  name: cas-add-mount
patch: |-
```

```
  - op: add
    path: /spec/controllerTemplate/spec/volumes/-
    value:
     name: mount-name
     nfs:
       path: /vol/path-to-be-mounted/
       server: host-or-server-where-path-is-located
  - op: add
    path: /spec/controllerTemplate/spec/containers/0/volumeMounts/-
    value:
      name: mount-name
      mountPath: "/path-to-be-mounted"
target:
  group: viya.sas.com
  kind: CASDeployment
  name: .*
  version: v1alpha1
```

3. Save and close the new .yaml file. Ensure that it is in the same directory as the kustomization.yaml file.

4. In the kustomization.yaml file, add the name of your new .yaml file in the transformers section:

   ```
   ...
   transformers:
   - bundles/deployment-type/overlays/required/transformers.yaml
   - mount-file-name.yaml
   ...
   ```

   If you are using the kustomization.yaml from Step 2 on page 11, uncomment the *mount-file-name*.yaml line and replace the variable with the actual file name.

5. Save and close the kustomization.yaml file.

Change accessMode

The default accessMode for the cas-default-data and cas-default-permstore persistentVolumeClaims is ReadWriteMany, because it is required for any backup controllers for CAS. It is not required for deployments with SMP CAS, but changing the access mode will make it difficult to move from SMP to MPP in the future. To change the access mode for either cas-default-data or cas-default-permstore, perform the following steps.

1. Create a new .yaml file with a name that reflects the action you want to perform. A good example is cas-access-mode.yaml, but you should use a name that is meaningful to you.

2. To modify the access mode for cas-default-data, add the following content:

   ```
   # access-file-name.yaml
   # Modify cas-default-data PVC access mode
   apiVersion: builtin
   kind: PatchTransformer
   metadata:
     name: pvc-access-mode
   patch: |-
    - op: replace
   ```

```
   path: /spec/accessModes
   value:
     - access-mode
target:
  kind: PersistentVolumeClaim
  name: cas-default-data
  version: v1
```

The *access-mode* can be ReadWriteMany or ReadWriteOnce.

3 To modify the access mode for cas-default-permstore, add the following content:

```
# access-file-name.yaml
# Modify cas-default-permstore PVC access mode
apiVersion: builtin
kind: PatchTransformer
metadata:
  name: pvc-access-mode
patch: |-
 - op: replace
   path: /spec/accessModes
   value:
     - access-mode
target:
  kind: PersistentVolumeClaim
  name: cas-default-permstore
  version: v1
```

If you want to modify the access mode for both cas-default-data and cas-default-permstore, put both code examples in your new .yaml file, separated by three dashes (`---`). You do not need to repeat the file name. Here is an example:

```
...
target:
  kind: PersistentVolumeClaim
  name: cas-default-data
  version: v1
---
# Modify cas-default-permstore PVC access mode
apiVersion: builtin
kind: PatchTransformer
...
```

4 Save and close the new .yaml file. Ensure that it is in the same directory as the kustomization.yaml file.

5 In the kustomization.yaml file, add the name of your new .yaml file in the transformers section:

```
...
transformers:
- bundles/deployment-type/overlays/required/transformers.yaml
- access-file-name.yaml
...
```

6 Save and close the kustomization.yaml file.

Adjust Default Resources for CAS Servers

If you want to adjust the default resources that are provided for your CAS servers, create a new .yaml file with a name that reflects the action you want to perform. A good example is cas-resource.yaml, but you should use a name that is meaningful to you.

1 Create a new .yaml file with a name that reflects the action you want to perform. A good example is cas-resource.yaml, but you should use a name that is meaningful to you.

2 The default resource allocation for RAM is 2 gigabytes. To modify the resources for memory usage, add the following content to the new .yaml file:

```
# resource-file-name.yaml
# Modify memory usage
apiVersion: builtin
kind: PatchTransformer
metadata:
  name: cas-memory-update
patch: |-
  - op: replace
    path: /spec/controllerTemplate/spec/containers/0/resources/requests/memory
    value:
      memory-resource-allocation
target:
  group: viya.sas.com
  kind: CASDeployment
  name: .*
  version: v1alpha1
```

The *memory-resource-allocation* should be a numeric value followed by the units, such as `3Gi`, for 3 gigabytes. The upper limit is 32GI.

Note: In Kubernetes, the units are Gi (gigabyte), Mi (megabyte) and Ki (kilobyte).

3 The default resource allocation for CPUs is .25 cores. To modify the resources for CPU usage, add the following content to the new .yaml file:

```
# resource-file-name.yaml
# Modify CPU usage
apiVersion: builtin
kind: PatchTransformer
metadata:
  name: cas-cpu-update
patch: |-
  - op: replace
    path: /spec/controllerTemplate/spec/containers/0/resources/requests/cpu
    value:
      CPU-resource-allocation
target:
  group: viya.sas.com
  kind: CASDeployment
  name: .*
  version: v1alpha1
```

The *CPU-resource-allocation* should either a whole number, representing that number of cores, or a number followed by `m`, indicating that number of milli-cores. So `8` would mean allocating 8 cores, and `5m` would mean allocating 500 milli-cores, or .5 cores. The upper limit is 8 cores.

4 The default resource allocation for ephemeral storage is 8 gigabytes. To modify the resources for ephemeral storage, add the following content to the new .yaml file:

```
# resource-file-name.yaml
# Modify ephemeral storage
apiVersion: builtin
kind: PatchTransformer
metadata:
  name: cas-ephemeral-storage-update
patch: |-
  - op: replace
    path: /spec/controllerTemplate/spec/containers/0/resources/requests/ephemeral-storage
    value:
      ephemeral-resource-allocation
target:
  group: viya.sas.com
  kind: CASDeployment
  name: .*
  version: v1alpha1
```

The *ephemeral-resource-allocation* should be a numeric value followed by the units, such as `16Gi`, for 16 gigabytes. The upper limit is 64Gi.

Note: In Kubernetes, the units are Gi (gigabyte), Mi (megabyte) and Ki (kilobyte).

```
...
target:
  group: viya.sas.com
  kind: CASDeployment
  name: .*
  version: v1alpha1
---
# Modify ephemeral storage
apiVersion: builtin
kind: PatchTransformer
...
```

5 Save and close the new .yaml file. Ensure that it is in the same directory as the kustomization.yaml file.

6 In the kustomization.yaml file, add the name of your new .yaml file in the transformers section:

```
...
transformers:
- bundles/deployment-type/overlays/required/transformers.yaml
- resource-file-name.yaml
...
```

7 Save and close the kustomization.yaml file.

Concatenate CAS Setting yaml Files

If you want to use a single file for all of your CAS setting customizations, you can do so. If you do, ensure that each section of code is separated by three dashes (`---`). Also ensure that you use the correct file name in the transformers section of the kustomization.yaml file.

Create the Kubernetes Manifests

On the kubectl machine, create the Kubernetes manifests by running Kustomize in the same directory as the kustomization.yaml.

```
kustomize build >base.yaml
```

The following message might be displayed:

```
well-defined vars that were never replaced:
SAS_COMPONENT_RELPATH_orchstrtncli,SAS_COMPONENT_TAG_orchstrtncli
```

If this message is displayed it can safely be ignored.

Deploy the Software

1 Deploy the software with the following command.

```
kubectl apply -n name-of-namespace -f base.yaml
```

After running the command, if you see the following messages or your pods are in CrashLoopBackOff for an hour or more, you may be experiencing a race condition. Rerun the command to resolve the race condition.

```
unable to recognize "base.yaml": no matches for kind "Pgcluster" in version
"crunchydata.com/v1"
unable to recognize "base.yaml": no matches for kind "CASDeployment" in version
"viya.sas.com/v1alpha1"
```

2 Wait for Kubernetes to create and start the pods. To determine if the pods have started:

```
kubectl -n name-of-namespace get pods
```

The output of this command looks like this:

```
NAMESPACE      NAME                                      READY   STATUS        RESTARTS   AGE
d10006         annotations-66dc4479fd-qfqqr              1/1     Running       0          5s
d10006         appregistry-bbbdfb78c-tcllv               1/1     Running       0          5s
d10006         audit-7c4ff4b8b8-zxg8k                    1/1     Running       0          5s
d10006         authorization-79d4f594b9-t9sbx            1/1     Running       0          5s
d10006         cachelocator-668fcdb544-hcxbs             1/1     Running       0          5s
d10006         cacheserver-7dc898d4bf-8dfgx              1/1     Running       0          5s
d10006         casaccessmanagement-64b5769d8f-mlmjf      1/1     Running       0          5s
d10006         casadministration-747746f94c-j2dm2        1/1     Running       0          5s
```

The value in the **Status** column is `Running` when the pods have been started.

Sign In as the sasboot User

Your SAS environment is deployed with an initial administrator account that is named sasboot. The password for this account has expired by default, so you must reset the password before you can sign in.

To reset the password:

1 Get the name of the pod that contains SASLogon and search for the characters, **`sasboot`**:

```
kubectl logs -n name-of-namespace $(kubectl -n name-of-namespace get pods | grep saslogon | cut -f1 -d' ') | grep sasboot
```

2 Sign in from a URL with this format:

   http://*name-of-namespace.ingress-host*/SASLogon/reset_password?code=password

3 Follow the instructions on the displayed web page to reset the password.

   If the URL has expired, restart the saslogon pod:

   ```
   kubectl -n name-of-namespace delete pod saslogon-pod-name
   ```

   Then go to the log and obtain the new URL. The URL expires 24 hours after the SAS Logon service restarts. For security purposes, the URL that is specified in a browser or in a text editor also expires, even if the password is not reset.

   After you reset the password, SAS Environment Manager automatically opens in your browser.

4 Click **Yes** for all of the assumable groups so that you have the permissions to perform subsequent tasks.

5 Share the following URL with any other users of your SAS Viya software so that they can access the deployment:

   http://*name-of-namespace.ingress-host*/SASDrive

Configure LDAP

Note: The tasks in this section are applicable if you deployed all your software. If you deployed the programming interface only, skip this section.

Configure the Connection to Your Identity Provider

After completing the installation of SAS Viya, you must configure the connection to your identity provider before your users can access SAS Environment Manager and SAS Visual Analytics.

While signed in as sasboot, configure the connection to your identity provider:

Note: Only LDAP-based identity providers are supported. You need to have basic familiarity with LDAP administration. For more information about the properties that are relevant for this procedure, see "sas.identities.providers.ldap" in *SAS Viya Administration: Configuration Properties*.

1. Select the 🔧 from the side menu to open the Configuration page.

2. On the Configuration page, select **Basic Services** from the list, and then select the **Identities service** from the list of services.

3. To configure user properties, in the **sas.identities.providers.ldap.user** section, click **New Configuration**:

   a. Specify a value for the **baseDN** required field. For the remaining fields, review the default values and make changes, as necessary. The default values are appropriate for most sites.

      Note: When using the LDAP protocol, passwords are transmitted over the network as clear-text. To secure the deployment, SAS recommends that you configure encrypted LDAP connections. For more information, see Encrypt LDAP Connections in *Encryption in SAS Viya: Data in Motion*.

      For each property that represents a user-level field in SAS, specify a corresponding property in the LDAP server software.

      TIP Consider specifying a custom filter to limit the user accounts that SAS Viya returns from your LDAP server.

   b. If you are performing a multi-tenant deployment and you are using a custom LDAP structure, select the **Apply configuration only to this tenant (provider)** option.

c Click **Save**.

4 To configure group properties, in the **sas.identities.providers.ldap.group** section, click **New Configuration**. In the New Configuration window:

a Specify a value for the **baseDN** required field. For the remaining fields, review the default values and make changes, as necessary. The default values are appropriate for most sites.

For each property that represents a group-level field in SAS, specify a corresponding property in the LDAP server software.

> **TIP** Consider specifying a custom filter to limit the accounts that SAS Viya returns from your LDAP server.

b If you are performing a multi-tenant deployment and you are using a custom LDAP structure, select the **Apply configuration only to this tenant (provider)** option.

If you are using the fixed LDAP structure that is described in , do not select this option.

c Click **Save**.

5 To configure connection properties, in the **sas.identities.providers.ldap.connection** section, click **New Configuration**. In the New Configuration window:

a Specify values for the following required fields: **host**, **password**, **port**, **url**, and **userDN**. For the remaining fields, review the default values and make changes, as necessary. The default values are appropriate for most sites.

b If you are not performing a multi-tenant deployment, skip this step.

If you are performing a multi-tenant deployment and you are using an LDAP server per tenant, select the **Apply configuration only to this tenant (provider)** option.

If you are using a single LDAP server for all tenants, or a fixed directory structure that applies to all tenants, do not select this option.

c Click **Save**.

6 To verify user and group information, from the SAS Environment Manager side menu, select 👥 to open the Users page.

On the Users page, select **Users** from the list in the toolbar. Your users should appear after a few minutes. It is not necessary to restart any servers or services. Then select **Groups** from the list to display your groups.

7 Verify that user and group information is displayed correctly. If not, make any necessary changes to the identities service properties, then restart the Identities and SAS Logon Manager services.

a Get the names of the pods that contain SASLogon and identities:

```
kubectl get pods  -n name-of-namespace | grep saslogon
```

```
kubectl get pods  -n name-of-namespace | grep identities
```

b Remove the SASLogon and identities pods:

```
kubectl -n name-of-namespace delete pod saslogon-pod-name
```

```
kubectl -n name-of-namespace delete pod identities-pod-name
```

Set Up Administrative Users

While you are signed on to SAS Environment Manager as the sasboot user, set up at least one SAS Administrator user, as follows:

1 On the Users page in SAS Environment Manager, select **Custom Groups** from the list in the toolbar.

2 In the left pane, click **SAS Administrators**.

3 In the **Members** section of the right pane, click ✎, and add one or more members to the group (including your own account, if applicable).

4 Sign out from SAS Environment Manager so that you are no longer signed in as the sasboot user.

5 If you added your own account to the SAS Administrators group, you can sign on again to SAS Environment Manager using that account.

Open SAS Environment Manager from a URL with the following format:

```
http://ingress-path/SASEnvironmentManager
```

> **TIP** Since SAS Administrators is an assumable group, the following prompt is displayed: `Do you want to opt in to all of your assumable groups?`. Select **Yes** if you want the extra permissions that are associated with the SAS Administrators group. The selection remains in effect until you sign out.

Sign In Using LDAP Credentials

Open SAS Environment Manager from a URL with the following format:

```
http://ingress-path/SASEnvironmentManager
```

Sign in as one of the SAS Administrators that you set up in “Set Up Administrative Users” on page 22.

5

Validating the Deployment

Verify SSSD 23

Verify SAS/ACCESS Interface to Oracle 23

Verify PC Files Connectivity 24

Verify SSSD

To verify that SSSD is set up correctly:

1 Find the pod ID for the compsrv pod:

```
kubectl -n name-of-namespace get pods | grep compsrv
```

The result will be the name of the pod for compsrv, such as compsrv-644697b6d5-pl2xn.

2 Test the SSSD configuration:

```
kubectl -n name-of-namespace exec -it compsrv-pod-ID -c compsrv getent passwd test-user
```

3 If SSSD is running as expected, the result looks something like this:

```
getent passwd sas
sas:x:1001:1001:SAS User account:/opt/sas/viya/home:/sbin/nologin
```

In this example of the output, “sas” is the *test-user* used in the command in step 2.

If the command returns nothing, SSSD is not configured correctly.

Verify SAS/ACCESS Interface to Oracle

From SAS Studio, edit and run the following SAS code to verify the SAS/ACCESS Interface to Oracle LIBNAME statement:

```
libname olib oracle path="path-to-database" user="" password="user-password";
```

If SAS/ACCESS Interface to Oracle was successfully deployed, the execution of the LIBNAME statement returns results without error.

Verify PC Files Connectivity

1 Log in to SAS Studio.

2 Run the following program:

```
/* Test PC files connectivity. */

/* Export from SAS to Excel file. */
proc export
  data=sashelp.prdsale
      dbms=xlsx
      outfile="prdsale.xlsx"
      replace;
run;

/* Read in Excel file. */
libname test xlsx 'prdsale.xlsx';

data prdsale;
  set test.prdsale;
run;
```

If the program runs successfully, the deployment has the appropriate connection to PC files.

6

Completing the Deployment

Administration Documentation 25

Administration Documentation

Now that your SAS software is deployed and running, see SAS Viya Administration: Initial Tasks for your initial administrative tasks.

7

Managing Your Software

Latest Requirements 27

Applying a New License for Your SAS Viya Software 27
Apply a New License 27

Updating Your SAS Viya Software 29
Retrieve Required Files 29
Configure Deployment Options 29
Deploy the Updated Software 30

Latest Requirements

The required versions of third-party software change over time. Always ensure that you have the latest deployment guide and meet the requirements described in it, even when working with an existing deployment of SAS Viya software.

Applying a New License for Your SAS Viya Software

Apply a New License

Your new license is available on the SAS Portal. The license file name will be similar to SASViyaV0400_XXXXXX_XXXXXXXX_Linux_x86-64.jwt.

1. To access your new license, click the **Get Started** link provided in your SOE.

2. On the My SAS web page that opens, expand the information for the order listed in your SOE by clicking the plus sign (“+”).

3. Under **Order Assets**, click **Download License**.

4. Save the license in the directory for your site edits.

5 Copy the renewal license file SASViyaV0400_XXXXXX_XXXXXXXX_Linux_x86-64.jwt to the base directory for this order.

6 Rename the license file from SASViyaV0400_XXXXXX_XXXXXXXX_Linux_x86-64.jwt to license.jwt.

Note: If license.jwt exists, overwrite it with the renamed SASViyaV0400_XXXXXX_XXXXXXXX_Linux_x86-64.jwt file.

7 If the `secretGenerator` block does not exist in kustomization.yaml, add the following code immediately before the configMapGenerator line:

```
secretGenerator:
 - name: sas-license
   type: sas.com/license
   behavior: merge
   files:
   - SAS_LICENSE=license.jwt
```

Note: If you used the default kustomization.yaml file described in "Create the kustomization.yaml File" on page 11, you only need to uncomment those same lines in your existing kustomization.yaml file.

8 If the `secretGenerator` block does exist in kustomization.yaml, and the set of lines after the `secretGenerator` line are either absent or different from what is shown, add the lines or edit the lines so the `secretGenerator` is exactly as shown.

Note: Do not delete or edit other lines in the `secretGenerator` if they are present.

9 On the kubectl machine, create the Kubernetes manifests by running Kustomize in the same directory as the kustomization.yaml.

```
kustomize build >base.yaml
```

The following message might be displayed:

```
well-defined vars that were never replaced:
SAS_COMPONENT_RELPATH_orchstrtncli,SAS_COMPONENT_TAG_orchstrtncli
```

If this message is displayed it can safely be ignored.

10 Deploy the software. See "Deploy the Software" on page 18.

Updating Your SAS Viya Software

Retrieve Required Files

To update your software, use the same steps to retrieve the required files that you used for the initial deployment. See “Retrieve Required Files” on page 9 for the specific steps.

Configure Deployment Options

1 Ensure that you are working from the directory where you saved the deployment assets in “Retrieve Required Files ” on page 29:

```
cd $deploy
```

2 Copy the kustomization.yaml file from */original-order-directory* and place it into */new-order-directory*. If you used a sitedefault.yml file from SAS Viya 3.X in your original deployment, also copy it to the same location where you placed the kustomization.yaml file for the new order.

3 On the kubectl machine, create the Kubernetes manifests by running Kustomize in the same directory as the kustomization.yaml and the updated deployment assets.

```
kustomize build >base.yaml
```

The following message might be displayed:

```
well-defined vars that were never replaced:
SAS_COMPONENT_RELPATH_orchstrtncli,SAS_COMPONENT_TAG_orchstrtncli
```

If this message is displayed it can safely be ignored.

4 Compare the **data/data_connectors.yaml** and **data/data_mounts.yaml** files from the original deployment with **data/data_connectors.sample.yaml** and **data/data_mounts.sample.yaml** in the new directory. If there are no structural changes in the new samples, such as new variables or old variables being removed, copy the **data/data_connectors.yaml** and **data/data_mounts.yaml** from the original deployment to the new directory. If there are structural differences, use the samples to create new files. Use content from the files from the original deployment to fill in variables that the original and new files have in common. Then fill in any new variables in the new files.

5 Compare **hacks/sssd/sas-sssd-configmap.yaml** from the original deployment with **hacks/sssd/sas-sssd-configmap.sample.yaml** in the new directory. If there are no structural changes in the new sample, such as new variables or old variables being removed, copy the **hacks/sssd/sas-sssd-configmap.yaml** from

the original deployment to the new directory. If there are structural differences, use the sample to create a new file. Use content from the file from the original deployment to fill in variables that the original and new files have in common. Then fill in any new variables in the new file.

6. Perform one of the following sets of steps, depending on if you have an internal or external instance of PostgreSQL. For more information about the types of instance, see “ Deployment Options” on page 7.

   - External:

     1. Copy the **`/default/overlays/external-postgres/external-postgres-transformer.yaml`** file from the original deployment to the same location in the directory for the update order.

     2. Copy the **`/default/overlays/external-postgres/kustomization.yaml`** file from the original deployment to the same location in the directory for the update order.

   - Internal:

     Compare the **`/default/overlays/crunchydata`** file in the original deployment to the same file in the update directory. Ensure that any modifications you made to the original file are repeated in the update version of the file.

7. Apply the kustomization files to the Kubernetes manifest. The command you use depends on whether you plan to source data using data connectors or external volumes.

   - No sourcing of data is used:

     ```
     ./kustomizer.sh --customer-like
     ```

   - Sourcing of data is used:

     ```
     export DATA="true"; ./kustomizer.sh --customer-like
     ```

   If you receive a message like the following, it can be safely ignored.

```
well-defined vars that were never replaced:
SAS_COMPONENT_RELPATH_orchstrtncli,SAS_COMPONENT_TAG_orchstrtncli
```

Deploy the Updated Software

Use the same steps to deploy the updated software that you used for the initial deployment. See “Deploy the Software” on page 18 for the specific steps.

Appendix 1

Using a Mirror Repository

Create a Mirror Repository 31
Add a Mirror Repository to kustomization.yaml 31

Create a Mirror Repository

Content goes here. Reference to Gitlab project or summary of Mirror Manager steps

Add a Mirror Repository to kustomization.yaml

1 Copy the mirror.yaml file from **bundles/*deployment-type*/examples/mirror/** to the same location as the kustomization.yaml file.

2 Open the mirror.yaml file and replace each instance of `MIRROR_HOST` with the fully qualified domain name (FQDN) of the mirror repository. When you have made all the replacements, save and close the file.

3 In the kustomization.yaml file, add mirror.yaml to the transformers section, immediately after the entry for bundles/*deployment-type*/overlays/required/transformers.yaml:

```
transformers:
- bundles/deployment-type/overlays/required/transformers.yaml
- mirror.yaml
configMapGenerator:
...
```

4 Add the following content to the kustomization.yaml file at the end of the configMapGenerator section:

```
...
- name: ccp-image-location
  behavior: merge
  literals:
  - CCP_IMAGE_PATH=FQDN-of-mirror-host
...
```