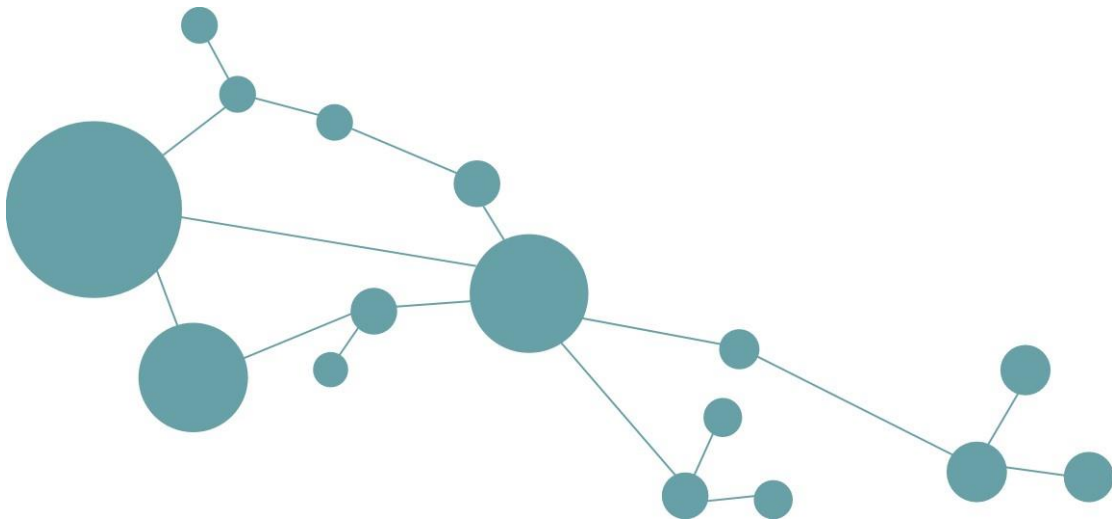




---

## Manuelle Installation von i-views



# Inhaltsverzeichnis

|          |                                                                 |           |
|----------|-----------------------------------------------------------------|-----------|
| <b>1</b> | <b>Allgemeine Information</b>                                   | <b>3</b>  |
| <b>2</b> | <b>Betriebssysteme</b>                                          | <b>5</b>  |
| 2.1      | Microsoft Windows . . . . .                                     | 5         |
| 2.2      | Unix-artige . . . . .                                           | 7         |
| 2.2.1    | Debian/Ubuntu (x86/amd64) . . . . .                             | 7         |
| 2.2.2    | RedHat/Fedora/CentOS (x64/amd64) . . . . .                      | 9         |
| 2.2.3    | Mac OS X 10.4+ (amd64) . . . . .                                | 10        |
| 2.2.4    | SUSE Linux Enterprise (SLES-12) . . . . .                       | 11        |
| 2.3      | Nicht offiziell unterstützte Betriebssysteme . . . . .          | 13        |
| 2.3.1    | FreeBSD . . . . .                                               | 13        |
| <b>3</b> | <b>Einrichten der Dienste</b>                                   | <b>13</b> |
| 3.1      | SystemV init . . . . .                                          | 14        |
| 3.2      | systemd . . . . .                                               | 14        |
| 3.3      | docker-compose . . . . .                                        | 16        |
| 3.4      | kubernetes . . . . .                                            | 17        |
| <b>4</b> | <b>Typische Anforderungen</b>                                   | <b>17</b> |
| 4.1      | Load-Balancing über mehrere gleichartige REST-Dienste . . . . . | 17        |
| 4.1.1    | Nginx . . . . .                                                 | 18        |
| 4.1.2    | Apache HTTPD . . . . .                                          | 18        |
| 4.1.3    | haproxy . . . . .                                               | 18        |
| 4.2      | Firewall-Konfiguration . . . . .                                | 19        |
| 4.2.1    | firewalld . . . . .                                             | 19        |
| <b>5</b> | <b>Anhang</b>                                                   | <b>19</b> |
| 5.1      | init.d-Skripte . . . . .                                        | 20        |
| 5.2      | docker-compose Konfiguration . . . . .                          | 34        |
| 5.3      | kubernetes Konfiguration . . . . .                              | 35        |



## 1 Allgemeine Information

Für den Betrieb von i-views ist es unerheblich, ob die Prozesse auf realer Hardware (Server, Workstation, Notebook, ...) oder in einer virtualisierten Umgebung (VMWare, VirtualBox, docker, ...) laufen.

Der Betrieb von i-views erfordert keine besonderen Berechtigungen innerhalb des Betriebssystems. Dadurch können die Prozesse der Produktteile von beliebigen Konten des Betriebssystems gestartet werden. Der Betrieb mit Systemkonten ist aber möglich, i-views arbeitet immer nur in den konfigurierten Datenbereichen und die Produktteile kommunizieren ausschließlich via TCP/IP.

Ohne weitere Konfiguration lesen und schreiben die Produktteile Daten und Log-Dateien in oder unterhalb ihres Arbeitsverzeichnis (d.h. dort sind Schreibrechte für das ausführende Konto notwendig).

Der Betrieb der Prozesse in einem nicht lokal verfügbaren Verzeichnis (z.B. einer Netzwerkfreigabe) wird nicht empfohlen, da dies abgesehen von schwacher I/O Performance in bestimmten Situationen zu Problemen führen kann.

Alle i-views-Produktteile laufen als eine Kombination von virtueller Maschine (VM) und sogenanntem Image. Die Image-Dateien enthalten den Code für den jeweiligen Produktteil und sind unabhängig vom Betriebssystem (und damit zwischen Betriebssystem austauschbar). Die mitgelieferten virtuellen Maschinen übernehmen die Abstraktion vom konkreten Betriebssystem. Für Windows bietet i-views Executables mit einer Kombination von VM und Image an.

Die virtuellen Maschinen und Images sind als 32-Bit- und 64-Bit-Varianten vorhanden. Die „Bittigkeit“ des Images erkennt man in Auslieferungszustand am (fehlenden) Zusatz „-64“ am Ende des Dateinamens des Images. Die „Bittigkeit“ des Images bestimmt die zu verwendende VM.

Es wird empfohlen - wann immer möglich - 32-Bit-Software einzusetzen. Unter einem 64-Bit-Betriebssystem müssen dazu teilweise zusätzliche Software-Pakete des Betriebssystems installiert werden (siehe unten). 64-Bit-Images sollten nur eingesetzt werden, wenn die 2GB Arbeitsspeicher eines 32-Bit-Prozesses nicht für die Bearbeitung der Aufgaben ausreichen (was üblicherweise auch bei großen Datenbeständen ausreicht).

Alle Produktteile kommunizieren untereinander via TCP/IP, so dass sie prinzipiell auch auf mehrere Maschinen verteilt werden können. Diese Verteilung ist nicht Teil dieses Dokuments. Falls die von den Diensten benutzten TCP/IP-Ports auch von außerhalb der jeweiligen Maschine erreicht werden sollen, müssen diese Ports über die Firewall-Mechanismen des jeweiligen Betriebssystems freigegeben werden.

Die Konfiguration der Produktteile findet über INI-Dateien statt. Einige Komponenten erwarten eine solche Datei, ohne die sie nur eine Fehlermeldung in die Log-Datei schreiben, bevor der Prozess beendet wird. Wird beim Aufruf keine INI-Datei explizit genannt, sucht der jeweilige Prozess im seinem Arbeitsverzeichnis nach einer INI-Datei, die zum Typ des Images passt (also z.B. mediator.ini für einen Mediator). Der Dateiname des Image oder Executables ist dabei unerheblich. Die individuelle Konfiguration der Produktteile ist nicht Teil dieses Dokuments.

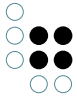
Alle Produktteile können über die Kommandozeile oder durch Start aus dem Dateimanager im Vordergrund gestartet werden. Die Prozesse werden beim Schließen des Fensters oder durch die Eingabe von Strg-C/Cmd-C beendet.

---



| <b>Komponente / Feature</b>    | <b>Beschreibung</b>                                                                                            | <b>Benötigt für...</b>                   |
|--------------------------------|----------------------------------------------------------------------------------------------------------------|------------------------------------------|
| Core 32 Bit                    | 32-Bit-Umgebung                                                                                                | Benötigt für Core                        |
| Core (De-)Kompression          | Kompressionswerkzeuge                                                                                          | Benötigt für Core                        |
| Core mit grafischer Oberfläche | i-views-Werkzeuge mit grafischer Oberfläche                                                                    | Benötigt für Core mit UI                 |
| Mediator-Prüfsummen            | Beschleunigte Berechnung von Prüfsummen für Knowledge Graph Inhalte                                            | Optional                                 |
| Bild-skalierung                | Beschleunigte Skalierung von Pixel-Bildern                                                                     | Optional                                 |
| Graph-Editor Alpha-Kanal       | Icons mit transparenten Bereichen im Graph Editor schöner anzeigen                                             | Optional für Core mit UI                 |
| JNI                            | Java Native Interface zur Anbindung von Java Software (z.B. Lucene)                                            | Benötigt für das Feature                 |
| Verschlüsselung                | Kryptographisch abgesicherte Verbindungen (z.B. HTTPS); Verschlüsselte Passwörter für Kommandozeilen-Werkzeuge | Benötigt für das Feature                 |
| Verbesserte Text Darstellung   | Texte mit besserem Anti-Aliasing und Kerning Anzeigen im UI                                                    | Optional für Core mit UI                 |
| MySQL-Anbindung                | Anbindung von MySQL Datenbanken                                                                                | Benötigt für das Feature                 |
| ODBC-Anbindung                 | Anbindung von ODBC Datenquellen                                                                                | Benötigt für das Feature                 |
| Oracle-Anbindung               | Anbindung von Oracle Datenbanken                                                                               | Benötigt für das Feature                 |
| SQLite-Anbindung               | Anbindung von SQLite Datenbanken                                                                               | Benötigt für das Feature und BlobService |
| Erweiterte Reguläre Ausdrücke  | Erweiterung der standardmäßig vorhandenen regulären Ausdrücke                                                  | Benötigt für das Feature                 |

Zu beachten ist, dass die zuvor erwähnte "Bittigkeit" der laufenden, virtuellen Maschine und deren Image der i-views Software externe Bibliotheken derselben "Bittigkeit" benötigen. Eine



Bibliothek, deren Bittigkeit nicht zur Software passt, wird nicht aktiviert werden. Dies ist auch der Fall, wenn eine der Bibliothek zugrundeliegende, referenzierte Bibliothek nicht in derselben Bittigkeit vorliegt.

## 2 Betriebssysteme

i-views unterstützt verschiedene Betriebssysteme. Dieses Kapitel beschreibt die Installation unter voll unterstützten Betriebssystemen. Am Ende finden sich Hinweise zur Installation unter weiteren möglichen Betriebssystemen, die aber nicht offiziell unterstützt werden.

Bei Bedarf können alle in i-views verwalteten Daten auf eine andere Systemumgebung (Hardware, Betriebssystem, ...) ohne Konvertierung migriert werden. Falls die Produktteile über mehrere Rechner verteilt betrieben werden, dürfen auch diese jeweils mit unterschiedlichen Betriebssystemen laufen (z.B. ein Mix von Windows und Linux). Auch für den Zugriff von Clients auf die Server gibt es keine Beschränkungen bei den verwendeten Betriebssystemen.

### 2.1 Microsoft Windows

i-views läuft auf allen bisher veröffentlichten Windows-Versionen mit NT Kernel (ab Windows 2000).

Unter Windows werden die Produktteile üblicherweise als EXE-Dateien ausgeliefert. Diese bestehen intern aus einer Kombination von virtueller Maschine und Image. Bei Bedarf können statt der EXE-Dateien auch unter Windows getrennte VM-Executables und Images verwendet werden.

Alle i-views-Produktteile, die den Betrieb als Dienst unterstützen, können unter Windows durch folgende (mit Administratorrechten gestartete) Kommandozeile als Dienst installiert werden:

```
PRODUKTTEIL.exe -installAsService DIENSTNAME [DIENST1 ...] [DIENSTPARAMETER ...]
```

Dieser Aufruf legt nur die Dienstkonfiguration an und beendet sich dann sofort wieder. Falls Dienste von anderen Diensten abhängen, können optional mehrere Dienstnamen **DIENST1** bis **DIENSTn** angegeben werden, die vorher von Windows gestartet sein sollen. Spezielle Parameter für den als Dienst laufenden Produktteil werden am Schluss angegeben.

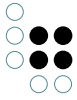
In der Windows-Dienst-Konfiguration (typischerweise [services.msc](#)) erscheint der Dienst danach unter dem Namen **DIENSTNAME** und kann dort bearbeitet werden, um z.B. das Konto, die Start-Art oder die Beschreibung anzupassen.

Die De-Registrierung erfolgt entweder über das Windows-Werkzeug „[sc delete DIENSTNAME](#)“ oder über:

```
PRODUKTTEIL.exe -deinstallService DIENSTNAME
```

Alternativ kann zur Dienst-Installation natürlich das Windows-eigene Werkzeug [sc.exe](#) eingesetzt werden.

| Funktion | Bibliothek | Zu installieren |
|----------|------------|-----------------|
|----------|------------|-----------------|



|                                |                                                      |                                                                                                                                                                                                                           |
|--------------------------------|------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Core 32 Bit                    | -                                                    | In Windows enthalten                                                                                                                                                                                                      |
| Core (De-)Kompression          | -                                                    | In Windows enthalten                                                                                                                                                                                                      |
| Core mit grafischer Oberfläche | -                                                    | In Windows enthalten                                                                                                                                                                                                      |
| Mediator-Prüfsummen            | coastbinary                                          | <a href="#">coastbinary.dll</a> : mitgeliefert, neben mediator.exe                                                                                                                                                        |
| Bildskalierung                 | GDIPlus                                              | In Windows enthalten                                                                                                                                                                                                      |
| Graph-Editor Alpha-Kanal       | Cairo                                                | <a href="#">libcairo-2.dll</a><br>Von <a href="http://www.gtk.org/download/windows.php">http://www.gtk.org/download/windows.php</a>                                                                                       |
| JNI                            | Java Virtual Maschine                                | <a href="#">libjvm.dll</a><br>wird bei der Installation einer Java Runtime (jre) oder eines Development Kits (jdk) mitgeliefert, z.B. unter <a href="http://java.com">java.com</a>                                        |
| Verschlüsselung                | Windows nativ und OpenSSL                            | <a href="#">bcrypt.dll</a> : in Windows enthalten<br><a href="#">libeay32.dll</a> : via <a href="https://wiki.openssl.org/index.php/Binaries">https://wiki.openssl.org/index.php/Binaries</a>                             |
| Verbesserte-Text Darstellung   |                                                      | In Windows enthalten                                                                                                                                                                                                      |
| MySQL-Anbindung                | MySQL Client Bibliotheken                            | <a href="#">libmysql.dll</a> : via <a href="https://dev.mysql.com/downloads/connector/c/">https://dev.mysql.com/downloads/connector/c/</a>                                                                                |
| ODBC-Anbindung                 | ODBC                                                 | In Windows enthalten bzw. als Feature nachinstallierbar                                                                                                                                                                   |
| Oracle-Anbindung               | Windows Advanced Services Oracle Client Bibliotheken | <a href="#">advapi32.dll</a> : Windows Feature Advanced Services, meist vorhanden.<br><a href="#">oci.dll</a> : Von <a href="http://www.oracle.com/">http://www.oracle.com/</a> die zur DB passende Version installieren. |
| SQLite-Anbindung               | SQLite                                               | <a href="#">sqlite3.dll</a><br>Von <a href="https://www.sqlite.org/download.html">https://www.sqlite.org/download.html</a>                                                                                                |



|                                               |             |                                                                                                                                                                                                   |
|-----------------------------------------------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Erweit-<br>erte<br>Reguläre<br>Aus-<br>drücke | Boost-Regex | <a href="http://www.boost.org/users/download/">boost_regex.dll, msvcp140.dll, vcruntime140.dll</a><br>Von <a href="http://www.boost.org/users/download/">http://www.boost.org/users/download/</a> |
|-----------------------------------------------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Unter Windows werden teilweise Bibliotheken der Microsoft Visual C++ Runtime benötigt. Je nach Art und Alter der Windows-Version, kann es notwendig sein, diese zusätzlich zu installieren.

Der empfohlene Weg ist, das Redistributable von der offiziellen Microsoft Seite Visual C++ Redistributable zu installieren. Hierbei sollte auf die zum Betriebssystem passende Sprache und auf die zur i-views Installation passende "Bittigkeit" (32Bit / 64Bit) geachtet werden. Auf diese Weise werden nicht nur über den Windows-Update Mechanismus Aktualisierungen zur Verfügung gestellt, sondern man vermeidet auch mehrfache Kopien der Bibliothek.

Alternativ kann man die Bibliotheken [msvcp140.dll](#) und/oder [vcruntime140.dll](#) auch zu den jeweiligen Executables in das Verzeichnis legen.

## 2.2 Unix-artige

i-views läuft unter verschiedenen Unix-artigen Betriebssystemen, wobei die Mehrzahl der Installationen unter diversen Varianten von Linux laufen.

Für i-views werden (noch) keine Repositories zur Installation und automatischem Upgrade der Software angeboten. Die Software wird aktuell als Archiv geliefert, das an den bevorzugten Ort kopiert werden kann.

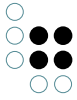
Die virtuellen Maschinen benötigen zum Start einige wenige Bibliotheken. Diese dürften auf den meisten Systemen out-of-the-box installiert sein, auf Minimalinstallationen können diese fehlen. Für bestimmte Features verwenden die i-views-Komponenten externe Bibliotheken, die dann aber erst bei Verwendung des Features angesprochen und nachgeladen werden. Falls die notwendigen Bibliotheken nicht gefunden werden, wird dieser Umstand in den Log-Dateien genannt. Hierbei kann es vorkommen, dass in den Log-Dateien die Haupt-Bibliothek als nicht verfügbar genannt wird, diese aber vorhanden ist, jedoch abhängige Bibliotheken fehlen. In diesem Fall hilft meist ein "[Idd BIBLIOTHEKSDATEI](#)" um vom System nicht gefundene abhängige Bibliotheken zu identifizieren.

Auf den meisten heute üblichen Servern wird per Default eine 64-Bit-Version eingesetzt. Um i-views als 32-Bit-Prozesse zu starten, ist es dann nötig, die 32-Bit-Laufzeitumgebung parallel zu installieren, wenn diese noch nicht vorhanden ist.

### 2.2.1 Debian/Ubuntu (x86/amd64)

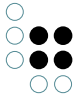
Debian/Ubuntu verwendet [apt](#) als Paketmanager. Es wird empfohlen, wo immer möglich Standard-Paket-Quellen zu verwenden, die aktuelle Patches liefern. Die Pakete werden mit "[apt-get install PAKETNAME](#)" installiert.

In der folgenden Tabelle sind die Paketnamen der Standard-Architektur erwähnt. Wird 32-Bit-i-views auf einer 64-Bit-Umgebung betrieben, muss bei der Installation das Suffix "[i386](#)" an den Paketnamen angehängt werden.



| Feature                                 | OS<br>Biblio-<br>thek                     | Paketname / Bibliothek                                                                                                                                                   |
|-----------------------------------------|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Core 32 Bit                             | GLibC<br>32 Bit                           | <a href="#">libc:i386</a>                                                                                                                                                |
| Core (De-<br>)Kompression               | Kom-<br>pres-<br>sion-<br>sutili-<br>ties | <a href="#">zlib1g</a>                                                                                                                                                   |
| Core mit<br>grafischer<br>Oberfläche    | X11<br>Umge-<br>bung                      | <a href="#">libx11-6</a>                                                                                                                                                 |
| Mediator<br>Prüfsum-<br>men             | coast-<br>bina-<br>ry.so                  | <a href="#">coastbinary.so</a><br>mitgeliefert, im Verzeichnis der VM                                                                                                    |
| Bild-<br>skalierung                     | FreeIm-<br>age                            | <a href="#">libfreeimage3</a>                                                                                                                                            |
| Graph-<br>Editor<br>Alpha-<br>Kanal     | Cairo                                     | <a href="#">libcairo2</a>                                                                                                                                                |
| JNI                                     | Java<br>Virtual<br>Ma-<br>chine           | <a href="#">libjvm.so</a><br>wird bei der Installation einer Java Runtime (jre) oder eines Devel-<br>opment Kits (jdk) mitgeliefert, z.B. unter <a href="#">java.com</a> |
| Verschlüs-<br>selung                    | OpenSSL                                   | <a href="#">libssl1.0.0</a>                                                                                                                                              |
| Verbesserte<br>Text<br>Darstel-<br>lung | XFT                                       | <a href="#">libxft2</a>                                                                                                                                                  |
| MySQL-<br>Anbindung                     | MySQL<br>oder<br>Mari-<br>aDB             | <a href="#">libmysqlclient20</a> oder <a href="#">libmariadb-client-lgpl-dev-compat</a>                                                                                  |
| ODBC-<br>Anbindung                      | iODBC                                     | <a href="#">libiodbc2</a>                                                                                                                                                |
| Oracle-<br>Anbindung                    | Oracle                                    | <a href="#">libclntsh.so</a><br>Von <a href="http://www.oracle.com/">http://www.oracle.com/</a> die zur DB passende Version instal-<br>lieren.                           |





|                               |             |                                      |
|-------------------------------|-------------|--------------------------------------|
| SQLite-Anbindung              | SQLite      | <a href="#">libsqlite3-0</a>         |
| Erweiterte Reguläre Ausdrücke | Boost-Regex | <a href="#">libboost-regex1.58.0</a> |

Unter Debian kann man Pakete, die ein bestimmtes Feature enthalten, mit "[apt-cache search 'NAME'](#)" finden.

Ob ein Paket wirklich die gewünschten Dateien enthält, kann man mit "[apt-file list 'PAKET'](#)" (aus dem Paket [apt-file](#)) überprüfen.

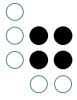
Alternativ kann eine fehlende Datei über die Webseiten der jeweiligen Distribution gefunden werden, also z.B. [https://www.debian.org/distrib/packages#search\\_contents](https://www.debian.org/distrib/packages#search_contents) bzw. <http://packages.ubuntu.com/> gesucht werden.

### 2.2.2 RedHat/Fedora/CentOS (x64/amd64)

RedHat-basierte Distributionen verwenden meist [yum](#) als Paketmanager. Es wird empfohlen, wo immer möglich Standard-Paket-Quellen zu verwenden, die aktuelle Patches liefern. Die Pakete werden mit "[yum install PAKETNAME](#)" installiert.

In der folgenden Tabelle sind die Paketnamen der Standard-Architektur erwähnt. Wird 32-Bit-i-views auf einer 64-Bit-Umgebung betrieben, muss bei der Installation das Suffix "[.i686](#)" an den Paketnamen angehängt werden.

| Feature                        | OS Bibliothek  | Paketname / Bibliothek                                                |
|--------------------------------|----------------|-----------------------------------------------------------------------|
| Core 32 Bit                    | GLibC 32 Bit   | <a href="#">glibc.i686</a>                                            |
| Core (De-)Kompression          | LibZ           | <a href="#">zlib</a>                                                  |
| Core mit grafischer Oberfläche | X11 Umgebung   | <a href="#">libX11</a>                                                |
| Mediator Prüfsummen            | coastbinary.so | <a href="#">coastbinary.so</a><br>mitgeliefert, im Verzeichnis der VM |
| Bildskalierung                 | FreeImage      | <a href="#">freeimage</a>                                             |



|                                     |                            |                                                                                                                                                                                                  |
|-------------------------------------|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Graph-Editor<br>Alpha-Kanal         | Cairo                      | cairo                                                                                                                                                                                            |
| JNI                                 | Java<br>Virtual<br>Machine | <a href="http://libjvm.so">libjvm.so</a><br>wird bei der Installation einer Java Runtime (jre) oder eines Development Kits (jdk) mitgeliefert, z.B. unter <a href="http://java.com">java.com</a> |
| Verschlüsselung                     | OpenSSL                    | <a href="http://openssl">openssl</a>                                                                                                                                                             |
| Verbesserte<br>Text<br>Darstellung  | XFT                        | <a href="http://libXft">libXft</a>                                                                                                                                                               |
| MySQL-Anbindung                     | Mariadb                    | <a href="http://mariadb-libs">mariadb-libs</a>                                                                                                                                                   |
| ODBC-Anbindung                      | iODBC                      | <a href="http://libiodbc">libiodbc</a>                                                                                                                                                           |
| Oracle-Anbindung                    | Oracle                     | <a href="http://libclntsh.so">libclntsh.so</a><br>Von <a href="http://www.oracle.com/">http://www.oracle.com/</a> die zur DB passende Version installieren.                                      |
| SQLite-Anbindung                    | SQLite                     | <a href="http://sqlite">sqlite</a>                                                                                                                                                               |
| Erweiterte<br>Reguläre<br>Ausdrücke | Boost-Regex                | <a href="http://boost-regex">boost-regex</a>                                                                                                                                                     |

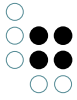
In RPM-basierten Distributionen kann man Pakete, die ein bestimmtes Feature enthalten mit "[yum search 'NAME'](#)" finden. Ob ein Paket wirklich die gewünschten Dateien enthält kann man mit "[repoquery -l 'PAKET'](#)" (aus dem Paket [yum-utils](#)) überprüfen.

Alternativ kann eine fehlende Datei über <https://rpmfind.net/linux/rpm2html/search.php> gesucht werden.

### 2.2.3 Mac OS X 10.4+ (amd64)

Einige Komponenten sind bereits in Mac OS X enthalten, andere kann man z.B. über MacPorts installieren, oder eben von den jeweiligen Projektseiten.

| Feature | OS Bibliothek | Name der Datei |
|---------|---------------|----------------|
|---------|---------------|----------------|

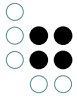


|                                |                      |                                                                                                        |
|--------------------------------|----------------------|--------------------------------------------------------------------------------------------------------|
| Core 32 Bit                    | LibC 32 Bit          | <a href="#">libc.dylib</a> , mit Mac OS X enthalten als <a href="#">libSystem.dylib</a>                |
| Core (De-)Kompression          | LibZ                 | In Mac OS X enthalten                                                                                  |
| Core mit grafischer Oberfläche | X11 Umgebung         | In Mac OS X enthalten                                                                                  |
| Mediator Prüfsummen            | coastbinary          | Prüfsummen werden intern berechnet, die externe Bibliothek gibt es nicht für Mac OS X                  |
| TLS connections                | tlsPlugin            | <a href="#">tlsPlugin.dylib</a> : mitgeliefert mit der Applikation                                     |
| Bildskalierung                 | Freelimage           | Verfügbar beim <a href="#">FreelImage Projekt</a>                                                      |
| Graph-Editor Alpha-Kanal       | Cairo                | <a href="#">libcairo.2.dylib</a> , via <a href="#">Cairo Project</a> oder <a href="#">MacPorts</a>     |
| JNI                            | Java Virtual Machine |                                                                                                        |
| Verschlüsselung                | OpenSSL              | <a href="#">libcrypto.1.0.dylib</a>                                                                    |
| Verbesserte Text Darstellung   | -                    | In Mac OS X enthalten                                                                                  |
| MySQL-Anbindung                | MySQL                | <a href="#">libmysqlclient.dylib</a> : von der <a href="#">MySQL</a> Webseite                          |
| ODBC-Anbindung                 | diverse, z.B. iODBC  | <a href="#">libiodbc.dylib</a> : z.B. von <a href="#">iodbc.org</a>                                    |
| Oracle-Anbindung               | Oracle               | <a href="#">libclntsh.dylib</a> : von <a href="#">Oracle</a> die zur DB passende Version installieren. |
| SQLite-Anbindung               | SQLite               | <a href="#">libsqlite3.dylib</a> : von <a href="#">sqlite.org</a>                                      |
| Erweiterte Reguläre Ausdrücke  | Boost-Regex          | <a href="#">libboost_regex.dylib</a> : von <a href="#">boost.org</a>                                   |

## 2.2.4 SUSE Linux Enterprise (SLES-12)

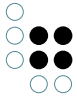
SUSE Distributionen verwenden meist [zypper](#) als Paketmanager. Es wird empfohlen, wo immer möglich Standard-Paket-Quellen zu verwenden, die aktuelle Patches liefern. Die Pakete werden mit "[zypper install PAKETNAME](#)" installiert.

In der folgenden Tabelle sind die Paketnamen der Standard-Architektur erwähnt. Wird 32-Bit-i-views auf einer 64-Bit-Umgebung betrieben, wird bei der Installation meist das Suffix



"-32bit" an den Paketnamen angehängt.

| Feature                                 | OS<br>Biblio-<br>thek           | Paketname / Bibliothek                                                                                                                                                   |
|-----------------------------------------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Core 32 Bit                             | GLibC<br>32 Bit                 | <a href="#">glibc-32bit</a>                                                                                                                                              |
| Core (De-<br>)Kompression               | LibZ                            | <a href="#">libz1</a>                                                                                                                                                    |
| Core mit<br>grafischer<br>Oberfläche    | X11<br>Umge-<br>bung            | <a href="#">libX11-6</a>                                                                                                                                                 |
| Mediator<br>Prüfsum-<br>men             | coast-<br>bina-<br>ry.so        | <a href="#">coastbinary.so</a><br>mitgeliefert, im Verzeichnis der VM                                                                                                    |
| Bild-<br>skalierung                     | FreeIm-<br>age                  | <a href="#">libfreeimage3</a>                                                                                                                                            |
| Graph-<br>Editor<br>Alpha-<br>Kanal     | Cairo                           | <a href="#">libcairo2</a>                                                                                                                                                |
| JNI                                     | Java<br>Virtual<br>Ma-<br>chine | <a href="#">libjvm.so</a><br>wird bei der Installation einer Java Runtime (jre) oder eines Devel-<br>opment Kits (jdk) mitgeliefert, z.B. unter <a href="#">java.com</a> |
| Verschlüs-<br>selung                    | OpenSSL                         | <a href="#">libopenssl1_0_0</a>                                                                                                                                          |
| Verbesserte<br>Text<br>Darstel-<br>lung | XFT                             | <a href="#">libXft2</a>                                                                                                                                                  |
| MySQL-<br>Anbindung                     | Mari-<br>aDB                    | <a href="#">libmysqlclient18</a>                                                                                                                                         |
| ODBC-<br>Anbindung                      | iODBC                           |                                                                                                                                                                          |
| Oracle-<br>Anbindung                    | Oracle                          | <a href="#">libclntsh.so</a><br>Von <a href="http://www.oracle.com/">http://www.oracle.com/</a> die zur DB passende Version instal-<br>lieren.                           |
| SQLite-<br>Anbindung                    | SQLite                          | <a href="#">libsqlite3-0</a>                                                                                                                                             |



|                                     |                 |                                      |
|-------------------------------------|-----------------|--------------------------------------|
| Erweiterte<br>Reguläre<br>Ausdrücke | Boost-<br>Regex | <a href="#">libboost_regex1_54_0</a> |
|-------------------------------------|-----------------|--------------------------------------|

## 2.3 Nicht offiziell unterstützte Betriebssysteme

Für jedes Betriebssystem wird eine spezifische virtuelle Maschine benötigt. Neben den oben beschriebenen Betriebssystemen gibt es VMs für:

- Mac OS X (PowerPC)
- Linux (SPARC), Linux (PPC)
- HPUX (PA RISC)
- AIX
- Solaris (SPARC), Solaris (x86+x86-64)

Diese VMs können auf Anfrage geliefert werden, es liegen aber wenige bis keine Erfahrungen zum Betrieb von i-views auf diesen Plattformen vor.

### 2.3.1 FreeBSD

FreeBSD bietet einen „Linux-binärkompatiblen Modus“ an. Details zu diesem Modus finden sich in der FreeBSD Dokumentation „10.2. Configuring Linux® Binary Compatibility“ unter <https://www.freebsd.org/doc/handbook/linuxemu-lbc-install.html>.

Nach Aktivierung des Modus kann man den Anweisungen auf der Webseite zu Linux folgen. Dabei sind zur Installation von möglicherweise noch nicht vorhandenen, aber benötigten Bibliotheken die FreeBSD-spezifischen Befehle zu verwenden.

## 3 Einrichten der Dienste

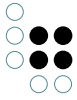
Das Starten der Produktteile hängt von der Ausführungsumgebung ab.

Für Linux Distributionen bestimmt das dort verwendete „init“-Systeme die notwendige Konfiguration. Für die i-views-Produktteile gibt es aktuell Beispieldateien für

- systemd-basierte Systeme wie CentOS7 und Ubuntu 16.04 und
- System-V-init.d-Skripte wie z.B. Suse-Enterprise-Linux-11-Skripte

Andere init-Systeme wie z.B. Upstart (wie CentOS 6.5) oder LaunchDaemon (Mac OS X) werden aktuell nicht angeboten, da sie bisher nur unvollständig oder nicht ausreichend generisch vorliegen. Da i-views nicht von einer speziellen Unterstützung durch das Betriebssystem abhängig ist, sollte es auf ähnlichen Plattformen einfach eingesetzt werden können.

Für Umgebungen, die auf Containern basieren, werden vom Build-System Standard Container-Images erstellt, die direkt verwendbar sein sollten. Hier werden Beispielfunktionen angeboten für:



- docker-compose
- kubernetes

Andere Container Laufzeit Umgebungen wie docker, podman oder openshift können von diesen Vorlagen abgeleitet werden.

### 3.1 SystemV init

Die init.d Skripte

- gehen von einer bash als ausführende Shell aus
- gehen von einer Installation in /opt/iviews/ aus
- versuchen aktuell, die Prozesse als Benutzer iviews zu kontrollieren.

Jedes Skript ist für sich selbst lauffähig, d.h. aber auch, dass man Umgebungsparameter üblicherweise in allen Skripten pflegen muss. Die Skripte und deren Parameter sind dokumentiert, im Zweifel sollten Parameter auf den dokumentierten Standard-Wert zurückgesetzt werden.

Die Skripte selbst sind im Anhangskapitel aufgeführt, da sie relativ lang sind.

### 3.2 systemd

Für den normalen Betrieb unter Distributionen, die systemd als init-System verwenden, können die folgenden Konfigurationsdateien als Vorlage verwendet werden:

[/etc/systemd/system/iviews-mediator.service:](#)

```
[Unit]
Description=iviews mediator
After=network.target nss-lookup.target

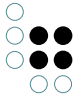
[Service]
User=iviews
WorkingDirectory=/opt/iviews/mediator
ExecStart=/opt/iviews/vw/vwlinux86 -noherald -=/opt/iviews/mediator/mediator.im
Restart=on-failure

[Install]
WantedBy=default.target
```

[/etc/systemd/system/iviews-bridge-rest.service:](#)

```
[Unit]
Description=iviews bridge rest
After=network.target nss-lookup.target iviews-mediator.service
Wants=iviews-mediator.service

[Service]
User=iviews
WorkingDirectory=/opt/iviews/bridge
```



```
ExecStart=/opt/iviews/vw/vwlinux86 -noherald --/opt/iviews/bridge/bridge.im -ini bridge-rest.ini  
Restart=on-failure
```

```
[Install]  
WantedBy=default.target
```

[/etc/systemd/system/iviews-jobclient.service:](#)

```
[Unit]  
Description=iviews jobclient  
After=network.target nss-lookup.target iviews-mediator.service  
Wants=iviews-mediator.service  
  
[Service]  
User=iviews  
WorkingDirectory=/opt/iviews/jobclient  
ExecStart=/opt/iviews/vw/vwlinux86 -noherald --/opt/iviews/jobclient/jobclient.im  
Restart=on-failure  
  
[Install]  
WantedBy=default.target
```

Sollen Dienste (Services) zu einer Installation zusammengefasst werden, z.B. wenn man mehrere Projekte parallel betreibt und diese getrennt starten und stoppen können will, ist es sinnvoll, ein projektspezifisches [target](#) zu definieren:

[/etc/systemd/system/iviews.target:](#)

```
[Unit]  
Description=iviews  
After=network.target nss-lookup.target  
Wants=iviews-mediator.service iviews-bridge-rest.service iviews-jobclient.service  
  
[Install]  
WantedBy=default.target
```

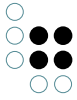
In den Dienst-Konfigurationen kann man dann die [\[Install\]](#) Sektionen entfernen und dafür in der [\[Unit\]](#) Sektion den Eintrag

```
PartOf=iviews.target
```

hinzufügen. Beim Starten und Stoppen der Unit [iviews.target](#) werden dann alle Teile gestartet bzw gestoppt, man kann aber auch Teildienste manuell neu starten, ohne das [iviews.target](#) zu beeinflussen.

Anpassen sind natürlich die Pfade der Verzeichnisse und Namen der Services an die konkrete Installation. Nach dem Editieren der Dateien muss man via [systemctl](#) die Dienste noch registrieren und starten.

Bei Bedarf können die Abhängigkeiten der Dienste auch stärker oder anders ausgedrückt werden. Details hierzu finden sich auf [systemd Homepage](#) oder natürlich in den man-pages der jeweiligen Distribution.



Cheat Sheet zur Verwaltung der Dienste:

| Aufgabe                                              | Kommando                                                                                              |
|------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| Einen Dienst registrieren                            | <code>systemctl enable SERVICE</code><br><br>z.B <code>sudo systemctl enable iviews-mediator</code>   |
| Einen Dienst deregistrieren                          | <code>systemctl disable SERVICE</code><br><br>z.B <code>sudo systemctl disable iviews-mediator</code> |
| Einen Dienst starten                                 | <code>systemctl start SERVICE</code><br><br>z.B <code>sudo systemctl start iviews-mediator</code>     |
| Einen Dienst beenden                                 | <code>systemctl stop SERVICE</code><br><br>z.B <code>sudo systemctl stop iviews-mediator</code>       |
| Den Dienststatus abfragen                            | <code>systemctl status SERVICE</code><br><br>z.B <code>sudo systemctl status iviews-mediator</code>   |
| Editieren einer Service Datei                        | <code>systemctl edit -full SERVICE</code>                                                             |
| Nach dem manuellen Editieren systemctl aktualisieren | <code>systemctl daemon-reload</code>                                                                  |

### 3.3 docker-compose

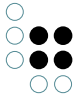
Das vorgestellte Beispiel bietet eine Konfigurationsdatei für docker-compose, die einen Mediator, eine Bridge und einen jobclient started. Es ist ebenfalls eine env-Datei enthalten, in der notwendige Kennwörter für die Installation hinterlegt sind. Das compose-file kann in Versionierungssystemen wie git gespeichert werden. Die env-Datei sollte im git keine echten Kennwörter enthalten.

In den Dateien sind Werte in eckigen Klammern (z.B. "<knowledge-graph-name>") enthalten, die als Platzhalter für Werte im Projekt dienen. Identische Platzhalter signalisieren identische Werte innerhalb der compose-Datei. Sie sind beabsichtigt nicht Teil der env-Datei, damit man nur eine solche verwenden muss (docker-compose verwendet im Standard ".env") und damit alle relevanten Angaben in einer Konfigurationsdatei enthalten sind.

Die Container-Images verwenden keine INI-Dateien, sondern erwarten die Konfigurationsangaben in Umgebungsvariablen mit IV\_ Präfix. Die i-views Werkzeuge unterstützen bei der Umwandlung existierender INI-Dateien zu Umgebungsvariablen, wenn sie mit dem Parameter -iniToEnv aufgerufen werden.

Die Konfigurationsdateien befinden sich im Anhangskapitel.





### 3.4 kubernetes

i-views kann in kubernetes Umgebungen ausgeführt werden. Der Mediator benötigt dazu einen zuverlässigen Storage-Treiber, nur die Speicherung auf Dateisystemen unterstützt wird.

In der Beispieldatei finden sich Werte in spitzen Klammern (z.B. "<knowledge-graph-name>"). Diese Platzhalter sind durch konkrete Werte des Projekts zu befüllen. Identische Platzhalter sollten mit dem selben Wert befüllt werden. Die Konfiguration mit einer Datei kann natürlich in mehrere yaml-Dateien aufgeteilt werden, wenn die Verwaltung der Installation mit einer Datei je Ressource stattfinden soll.

Zu beachten:

- Fast alle genannten Werte in der Konfiguration sind abhängig vom Projekt, der kubernetes Umgebung und Vorgaben für diese.
- Speicher per NFS bereit zu stellen hat sich bisher als sehr unzuverlässig erwiesen.
- Prozesse mit permanentem Speicher (insbesondere der Mediator) sollten nicht zwischen Knoten verschoben werden (siehe PodDisruptionBudget, maxUnavailable: 0).

Die Konfigurationsdatei findet sich im Anhangskapitel.

## 4 Typische Anforderungen

### 4.1 Load-Balancing über mehrere gleichartige REST-Dienste

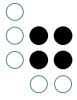
Bei REST-Services bietet es sich an, durch mehrere Diensterbringer einen höheren Durchsatz an parallelen Requests zu erreichen. i-views selbst ermöglicht mit einer Bridge mit LoadBalancer-Konfiguration das Starten und Überwachen mehrerer REST-Bridges. In der Konfigurationsdatei der Bridge werden dann z.B. folgende Einstellungen vorgenommen:

```
[KLoadBalancer]
hostname=localhost
port=5001
vm=/opt/kinfinity/vm/vwlinux86
directory=.
image=bridge.im
configNames=REST
autoRestart=true

[REST]
bridgeClientClassName=KWeb.KHTTPRestBridge
inifile=bridge_rest.ini
bridgeLogfile=bridge_rest_<id>.log
ports=5002-5005
```

Diese Konfiguration startet vier REST-Bridges auf den Ports 5001 bis 5005.

Allerdings unterstützt diese LoadBalancer-Bridge nicht selbst das LoadBalancing. Hierfür werden üblicherweise externe Pakete wie Nginx oder Apache-HTTPD eingesetzt.



#### 4.1.1 Nginx

Nginx kann mit der [proxy\\_pass](#) Direktive Anfragen an einen bestimmten Pfad intern an verschiedene Backends weiterleiten. Dazu erweitert man die Konfiguration von nginx um folgende Zeilen:

```
upstream kinf-rest-bridges {
    server 127.0.0.1:5002;
    server 127.0.0.1:5003;
    server 127.0.0.1:5004;
    server 127.0.0.1:5005;
}

server {
    ...

    location ^ ~ /myRestPath {
        proxy_pass http://kinf-rest-bridges/myRestPath;
    }
}
```

#### 4.1.2 Apache HTTPD

Apache HTTPD enthält typischerweise das Modul Proxy, welches u.a. Load Balancing anbietet. Eine Konfiguration für obige Bridge könnte folgendermaßen aussehen:

```
<Proxy balancer://kinfinity>
    BalancerMember http://127.0.0.1:5002/ disablereuse=On
    BalancerMember http://127.0.0.1:5003/ disablereuse=On
    BalancerMember http://127.0.0.1:5004/ disablereuse=On
    BalancerMember http://127.0.0.1:5005/ disablereuse=On
</Proxy>
ProxyPass /myRestPath balancer://kinfinity lbmethod=bybusyness
```

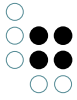
#### 4.1.3 haproxy

haproxy ist im Gegensatz zu nginx oder apache nur ein reiner Proxy-Dienst. haproxy wird typischerweise konfiguriert in

[/etc/haproxy/haproxy.cfg](#):

```
frontend          main *:5000
default_backend   rest_bridge

backend rest_bridge
    balance first
    server rest_bridge1 127.0.0.1:5002 check maxconn 1 weight 100
    server rest_bridge2 127.0.0.1:5003 check maxconn 1 weight 99
    server rest_bridge3 127.0.0.1:5004 check maxconn 1 weight 98
```



```
server rest_bridge4 127.0.0.1:5005 check maxconn 1 weight 97
```

## 4.2 Firewall-Konfiguration

i-views-Prozesse kommunizieren untereinander ausschließlich über TCP/IP. Die konkret verwendeten Ports hängen von der eingesetzten Software-Version und natürlich der lokalen Konfiguration ab.

Je nach eingesetztem OS und Distribution findet die Konfiguration unterschiedlich statt, daher kann hier keine komplette Dokumentation aller Firewall-Utilities vorgestellt werden. Z.B. unter Linux gibt es diverse Frontends und Konfigurationsweisen, welche aber alle die unterliegende iptables- bzw. nftables-Implementierung im Kernel steuern.

### 4.2.1 firewalld

firewalld, wie es z.B. unter CentOS7 eingesetzt wird, verwendet zur Konfiguration Zonen und Dienste.

Firewalld verwendet XML-Konfigurationsdateien, in denen man z.B. für eine Installation die komplette Konfiguration der Firewall-Regeln in einer Datei eintragen kann. Die Dateien werden in `/etc/firewalld/services/NAME.xml` abgelegt.

[/etc/firewalld/services/kinfinity.xml](#):

```
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>iviews-example</short>
  <description>i-views example installation</description>
  <!-- mediator -->
  <port protocol="tcp" port="30061"/>
  <!-- rest-bridge -->
  <port protocol="tcp" port="8815"/>
</service>
```

Eine neue Konfiguration kann man dann dauerhaft (`-permanent`) mit folgenden Befehlen für die öffentliche Zone (`public`) aktivieren:

```
firewall-cmd -permanent -zone=public -add-service=kinfinity
firewall-cmd -reload
```

Alternativ kann man auch die entsprechende Zonendatei manuell erweitern.

[/etc/firewalld/zones/public.xml](#):

```
<service name="kinfinity"/>
```

## 5 Anhang



## 5.1 init.d-Skripte

[/etc/init.d/mediator:](#)

```
#!/bin/bash
#
# Mediator Control-Script
#
#   chkconfig: 2345 20 80
#   description: Startup/shutdown script for Mediator

#=====
#== change the following parameters up to the second double line
#== to fit to the local installation

#-----
#- environment specific parameters

# path to external programs used in this script

# sudo, to support running as different user
SUDO=/usr/bin/sudo

# id to test current user (supporting options -nu)
ID=/usr/bin/id

# rm to delete pid files
RM=/bin/rm

# sh for execution of sudo commands in a subshell
SH=/bin/bash

#-----
#- mediator specific parameters

# directory, where the mediator will run in
# volumes will be expected in the 'volumes' subdirectory, some logfiles are dropped there too
#   (format: path, default: ".")
DIRECTORY="/opt/iviews/mediator"

# what image should be used for the mediator
#   (format: path+file, default: "$DIRECTORY/mediator.im")
IMAGE=""

# OPTIONAL: extra mediator parameters
#   (format: "(-OPTION)*", default: none)
PARAMS=""

# OPTIONAL: stop method: KILL|SHUTDOWN
# see documentation for details on SHUTDOWN
#   (format: "KILL|SHUTDOWN", default: "SHUTDOWN")
STOPMETHOD=""
```



```
# OPTIONAL: user to run as
# (format: username, default: none)
RUNASUSER="iviews"

#-----
#- mediator parameters, possibly defined in the ini-file

# port, where the mediator should run on
# (format: number, default: none)
PORT=""

# OPTIONAL: mediator logfile location
# (format: path+file, default: "$DIRECTORY/mediator.log")
LOGFILE=""

#-----
#- script specific parameters

# OPTIONAL: pidfile location (unused for stop, only used by kill)
# (format: path+file, default: "$DIRECTORY/mediator.pid")
PIDFILE=""

#-----
#- visualworks specific parameters

# where is the vm for that image located
# (format: path+file, default: "$DIRECTORY/vwlinux86")
VM=""
# note: on some platforms it might be necessary to prefix the vm
#       command by "setarch i386 -L " or similar, when normal starts
#       directly lead to a segmentation fault

# vm parameters ("-noherald" is usually a good idea for services)
# (format: "(-OPTION)*", default: none)
VM_PARAMS="-noherald"

#=====
#== no need to change anything below

PRODUCT="mediator"

#-----
#- parameter checking and defaulting

if test -z "$DIRECTORY"; then DIRECTORY='.'; fi
if test ! -d "$DIRECTORY" -o ! -w "$DIRECTORY"; then
    echo "directory '$DIRECTORY' not accessible, abort"; exit 1
fi

if test -z "$VM"; then VM="$DIRECTORY/vwlinux86"; fi
if test ! -f "$VM" -o ! -x "$VM"; then
    echo "virtual machine '$VM' not accessible, abort"; exit 1
fi
```



```
if test -z "$IMAGE"; then IMAGE="$DIRECTORY/$PRODUCT.im"; fi
if test ! -f "$IMAGE" -o ! -r "$IMAGE"; then
    echo "image '$IMAGE' not accessible, abort"; exit 1
fi

if test -n "$PORT"; then PARAMS="$PARAMS -port $PORT"; fi

if test -n "$LOGFILE"; then PARAMS="$PARAMS -log $LOGFILE"; fi

if test -z "$PIDFILE"; then PIDFILE="$DIRECTORY/$PRODUCT.pid"; fi

#-----
#- actions

startproc () {
    if test $# -lt 0; then return; fi
    local CMD=$1
    local BGFG=$2
    local PIDFILE=$3
    if test -z "$CMD"; then return; fi

    CMD="$CMD <&- >&- 2>&-"
    if test "$BGFG" = bg; then
        CMD="$CMD & disown"
        if test -n "$PIDFILE"; then
            CMD="$CMD; echo \#! > '$PIDFILE'"
        fi
    fi
fi

if test -z "$RUNASUSER" -o "$RUNASUSER" = $($ID -nu); then
    eval "$CMD"
else
    $SUDO -u "$RUNASUSER" $SH -c "$CMD"
fi
}

case "$1" in
start)
    echo -n "Starting k-infinity $PRODUCT in $DIRECTORY: "
    cwd=$(pwd); cd $DIRECTORY
    startproc "'$VM' $VM_PARAMS '$IMAGE' $PARAMS" bg "$PIDFILE"
    cd "$cwd"
    echo "done"
    ;;
stop)
    echo -n "Stopping k-infinity $PRODUCT in $DIRECTORY: "
    cwd=$(pwd); cd $DIRECTORY
    startproc "'$VM' $VM_PARAMS '$IMAGE' $PARAMS -stop" fg
    startproc "$RM -f '$PIDFILE'" fg
    cd "$cwd"
    echo "done"
    ;;
;
```



```
kill)
    if test -n "$PIDFILE"; then
        echo -n "Killing k-infinity $PRODUCT in $DIRECTORY: "
        if test ! -f "$PIDFILE" -o ! -r "$PIDFILE";
            then echo "missing pidfile '$PIDFILE', abort"; exit 1
            else
                startproc "kill -KILL \$(< '$PIDFILE')" fg
                startproc "$RM -f '$PIDFILE'" fg
            fi
        echo 'done'
        echo 'Do not forget to clean up all volume directories'
    else echo 'No PIDFILE parameter specified, NOT killed!'
    fi
    ;;
restart)
    $0 stop; $0 start;
    ;;
*)
    echo "Usage: $0 {start|stop|kill|restart}"
    exit 1
    ;;
esac

#=====
#== the end

exit 0
```

[/etc/init.d/bridge:](#)

```
#!/bin/bash
#
# Bridge Control-Script
#
#   chkconfig: 2345 30 70
#   description: Startup/shutdown script for Bridge

#=====
#== change the following parameters up to the second double line
#== to fit to the local installation

#-----
#- environment specific parameters

# path to external programs used in this script

# sudo, to support running as different user
SUDO=/usr/bin/sudo

# id to test current user (supporting options -nu)
ID=/usr/bin/id

# rm to delete pid files
```



```
RM=/bin/rm

# sh for execution of sudo commands in a subshell
SH=/bin/bash

#-----
#- bridge specific parameters

# directory, where the bridge will run in
# (format: path, default: ".")
DIRECTORY="/opt/iviews/bridge"

# what image should be used for the bridge in
# (format: path+file, default: "$DIRECTORY/bridge.im")
IMAGE=""

# OPTIONAL: ini-file location
# (format: path+file, default: none)
INIFILE=""

# OPTIONAL: extra parameters for the bridge
# (format: "(-OPTION)*", default: none)
PARAMS=""

# OPTIONAL: stop method: KILL|SHUTDOWN
# see documentation for details on SHUTDOWN
# (format: "KILL|SHUTDOWN", default: "SHUTDOWN")
STOPMETHOD=""

# OPTIONAL: user to run as
# (format: username, default: none)
RUNASUSER="iviews"

#-----
#- bridge parameters, usually better defined in the ini-file

# OPTIONAL: the mediator location, where to connect to
# (format: "HOST[:PORT]", default: none)
MEDIATOR=""

# OPTIONAL: local port to start on
# (format: "PORT", default: none)
LOCALPORT=""

#-----
#- script specific parameters

# pidfile location. used for stop with KILL method
# (format: path+file, default: "$DIRECTORY/bridge.pid")
PIDFILE=""

# The maximum time in seconds that a regular service shutdown may take
# (time from the execution of the shutdown command to the termination of the
```





```
# process). If the process is still running after the given time, it will
# be terminated by a kill command. Setting the timeout to a zero or negative
# value will disable that mechanism. The timeout parameter will only be
# evaluated if STOPMETHOD is set to "SHUTDOWN" and if PIDFILE is set to
# a valid path.
TIMEOUT=30
```

```
#-----
#- visualworks specific parameters
```

```
# where is the vm for that image located
# (format: path+file, default: "$DIRECTORY/vwlinux86")
VM=""
# note: on some platforms it might be necessary to prefix the vm
#       command by "setarch i386 -L " or similar, when normal starts
#       directly lead to a segmentation fault
```

```
# vm parameters ("-noherald" is usually a good idea for services)
# (format: "(-OPTION)*", default: none)
VM_PARAMS="-noherald"
```

```
#=====
#== no need to change anything below
```

```
PRODUCT="bridge"
```

```
#-----
#- parameter checking and defaulting
```

```
if test -z "$DIRECTORY"; then DIRECTORY='.'; fi
if test ! -d "$DIRECTORY" -o ! -w "$DIRECTORY"; then
    echo "directory '$DIRECTORY' not accessible, abort"; exit 1
fi
```

```
if test -z "$VM"; then VM="$DIRECTORY/vwlinux86"; fi
if test ! -f "$VM" -o ! -x "$VM"; then
    echo "virtual machine '$VM' not accessible, abort"; exit 1
fi
```

```
if test -z "$IMAGE"; then IMAGE="$DIRECTORY/$PRODUCT.im"; fi
if test ! -f "$IMAGE" -o ! -r "$IMAGE"; then
    echo "image '$IMAGE' not accessible, abort"; exit 1
fi
```

```
if test -z "$PIDFILE"; then PIDFILE="$DIRECTORY/$PRODUCT.pid"; fi
```

```
if test -n "$INIFILE"; then PARAMS="$PARAMS -ini $INIFILE"; fi
```

```
if test -n "$MEDIATOR"; then PARAMS="$PARAMS -host $MEDIATOR"; fi
```

```
if test -n "$LOCALPORT"; then PARAMS="$PARAMS -port $LOCALPORT"; fi
```

```
#-----
```



```
#- actions

startproc () {
    if test $# -lt 0; then return; fi
    local CMD=$1
    local BGFG=$2
    local PIDFILE=$3
    if test -z "$CMD"; then return; fi

    CMD="$CMD <&- >&- 2>&-"
    if test "$BGFG" = bg; then
        CMD="$CMD & disown"
        if test -n "$PIDFILE"; then
            CMD="$CMD; echo \#! > '$PIDFILE'"
        fi
    fi
    fi

    if test -z "$RUNASUSER" -o "$RUNASUSER" = $($ID -nu); then
        eval "$CMD"
    else
        $SUDO -u "$RUNASUSER" $SH -c "$CMD"
    fi
}

case "$1" in
    start)
        echo -n "Starting k-infinity $PRODUCT in $DIRECTORY: "
        cwd=$(pwd); cd $DIRECTORY
        startproc "'$VM' $VM_PARAMS '$IMAGE' $PARAMS" bg "$PIDFILE"
        cd "$cwd"
        echo "done"
        ;;
    stop)
        echo -n "Stopping k-infinity $PRODUCT in $DIRECTORY: "
        if test "$STOPMETHOD" = KILL; then
            if test ! -f "$PIDFILE" -o ! -r "$PIDFILE";
            then echo "missing pidfile '$PIDFILE', abort"; exit 1
            else
                startproc "kill -KILL \$(< '$PIDFILE')" fg
                startproc "$RM -f '$PIDFILE'" fg
            fi
        else
            cwd=$(pwd); cd $DIRECTORY
            startproc "'$VM' $VM_PARAMS '$IMAGE' $PARAMS -stop localhost" fg
            if test -f "$PIDFILE" ; then
                if test "$TIMEOUT" -gt 0 ; then
                    PID=$(cat $PIDFILE)
                    for (( i=0 ; i < $TIMEOUT ; i++ )) do
                        if ps -p $PID /dev/null 2>&1 ; then
                            sleep 1s
                            echo -n "."
                        else
                            startproc "$RM -f '$PIDFILE'" fg
                        fi
                    done
                fi
            fi
        fi
    esac
```



```
        i=$TIMEOUT;
        fi
    done
    if [ -f "$PIDFILE" ]; then
        echo -n " regular shutdown failed - killing process! "
        startproc "kill -KILL \$(< '$PIDFILE')" fg
        startproc "$RM -f '$PIDFILE'" fg
    fi
    else
        startproc "$RM -f '$PIDFILE'" fg
    fi
    fi
    cd "$cwd"
    fi
    echo "done"
    ;;
kill)
    echo -n "Killing k-infinity $PRODUCT in $DIRECTORY: "
    if test ! -f "$PIDFILE" -o ! -r "$PIDFILE";
    then echo "missing pidfile '$PIDFILE', abort"; exit 1
    else
        startproc "kill -KILL \$(< '$PIDFILE')" fg
        startproc "$RM -f '$PIDFILE'" fg
    fi
    echo "done"
    ;;
restart)
    $0 stop; $0 start;
    ;;
*)
    echo "Usage: $0 {start|stop|kill|restart}"
    exit 1
    ;;
esac

#=====
#== the end

exit 0

/etc/init.d/blobservice:

#!/bin/bash
#
# Blobservice Control-Script
#
#   chkconfig: 2345 30 70
#   description: Startup/shutdown script for Blobservice

#=====
#== change the following parameters up to the second double line
#== to fit to the local installation
```



```
#-----  
#- environment specific parameters  
  
# path to external programs used in this script  
  
# sudo, to support running as different user  
SUDO=/usr/bin/sudo  
  
# id to test current user (supporting options -nu)  
ID=/usr/bin/id  
  
# rm to delete pid files  
RM=/bin/rm  
  
# sh for execution of sudo commands in a subshell  
SH=/bin/bash  
  
#-----  
#- blobservice specific parameters  
  
# directory, where the blobservice will run in  
# (format: path, default: ".")  
DIRECTORY="/opt/iviews/blobservice"  
  
# what image should be used for the blobservice in  
# (format: path+file, default: "$DIRECTORY/blobservice.im")  
IMAGE=""  
  
# OPTIONAL: ini-file location  
# (format: path+file, default: none)  
INIFILE=""  
  
# OPTIONAL: extra parameters for the blobservice  
# (format: "(-OPTION)*", default: none)  
PARAMS=""  
  
# OPTIONAL: user to run as  
# OPTIONAL: user to run as  
# (format: username, default: none)  
RUNASUSER="iviews"  
  
#-----  
#- blobservice parameters, usually better defined in the ini-file  
  
# OPTIONAL: the bridge location, where to connect to  
# (format: "HOST[:PORT]", default: none)  
MEDIATOR=""  
  
# OPTIONAL: local port to start on  
# (format: "PORT", default: none)  
LOCALPORT=""  
  
#-----
```



```
#- script specific parameters

# pidfile location. used for stop with KILL method
# (format: path+file, default: "$DIRECTORY/blobservice.pid")
PIDFILE=""

#-----
#- visualworks specific parameters

# where is the vm for that image located
# (format: path+file, default: "$DIRECTORY/vwlinux86")
VM=""
# note: on some platforms it might be necessary to prefix the vm
#       command by "setarch i386 -L " or similar, when normal starts
#       directly lead to a segmentation fault

# vm parameters ("-noherald" is usually a good idea for services)
# (format: "(-OPTION)*", default: none)
VM_PARAMS="-noherald"

#=====
#== no need to change anything below

PRODUCT="blobservice"

#-----
#- parameter checking and defaulting

if test -z "$DIRECTORY"; then DIRECTORY='.'; fi
if test ! -d "$DIRECTORY" -o ! -w "$DIRECTORY"; then
    echo "directory '$DIRECTORY' not accessible, abort"; exit 1
fi

if test -z "$VM"; then VM="$DIRECTORY/vwlinux86"; fi
if test ! -f "$VM" -o ! -x "$VM"; then
    echo "virtual machine '$VM' not accessible, abort"; exit 1
fi

if test -z "$IMAGE"; then IMAGE="$DIRECTORY/$PRODUCT.im"; fi
if test ! -f "$IMAGE" -o ! -r "$IMAGE"; then
    echo "image '$IMAGE' not accessible, abort"; exit 1
fi

if test -z "$PIDFILE"; then PIDFILE="$DIRECTORY/$PRODUCT.pid"; fi

if test -n "$INIFILE"; then PARAMS="$PARAMS -ini $INIFILE"; fi

if test -n "$MEDIATOR"; then PARAMS="$PARAMS -host $MEDIATOR"; fi

if test -n "$LOCALPORT"; then PARAMS="$PARAMS -port $LOCALPORT"; fi

#-----
#- actions
```



```
startproc () {
    if test $# -lt 0; then return; fi
    local CMD=$1
    local BGFG=$2
    local PIDFILE=$3
    if test -z "$CMD"; then return; fi

    CMD="$CMD <&- >&- 2>&-"
    if test "$BGFG" = bg; then
        CMD="$CMD & disown"
        if test -n "$PIDFILE"; then
            CMD="$CMD; echo \#! > '$PIDFILE'"
        fi
    fi
    fi

    if test -z "$RUNASUSER" -o "$RUNASUSER" = "$($ID -nu); then
        eval "$CMD"
    else
        $SUDO -u "$RUNASUSER" $SH -c "$CMD"
    fi
}

case "$1" in
    start)
        echo -n "Starting k-infinity $PRODUCT in $DIRECTORY: "
        cwd=$(pwd); cd $DIRECTORY
        startproc "'$VM' $VM_PARAMS '$IMAGE' $PARAMS" bg "$PIDFILE"
        cd "$cwd"
        echo "done"
        ;;
    stop)
        echo -n "Stopping k-infinity $PRODUCT in $DIRECTORY: "
        if test ! -f "$PIDFILE" -o ! -r "$PIDFILE";
            then echo "missing pidfile '$PIDFILE', abort"; exit 1
        else
            startproc "kill -KILL \$(< '$PIDFILE')" fg
            startproc "$RM -f '$PIDFILE'" fg
        fi
        echo "done"
        ;;
    restart)
        $0 stop; $0 start;
        ;;
    *)
        echo "Usage: $0 {start|stop|restart}"
        exit 1
        ;;
esac

#=====
#== the end
```



```
exit 0
```

[/etc/init.d/jobclient:](#)

```
#!/bin/bash
#
# JobClient Control-Script
#
#   chkconfig: 2345 30 70
#   description: Startup/shutdown script for JobClient

#=====
#== change the following parameters up to the second double line
#== to fit to the local installation

#-----
#- environment specific parameters

# path to external programs used in this script

# sudo, to support running as different user
SUDO=/usr/bin/sudo

# id to test current user (supporting options -nu)
ID=/usr/bin/id

# rm to delete pid files
RM=/bin/rm

# sh for execution of sudo commands in a subshell
SH=/bin/bash

#-----
#- jobclient specific parameters

# directory, where the jobclient will run in
#   (format: path, default: ".")
DIRECTORY="/opt/iviews/jobclient"

# what image should be used for the jobclient in
#   (format: path+file, default: "$DIRECTORY/jobclient.im")
IMAGE=""

# MANDATORY: ini-file location
#   (format: path+file, default: "$DIRECTORY/jobclient.ini")
INIFILE=""

# OPTIONAL: extra parameters for the jobclient
#   (format: "(-OPTION)*", default: none)
PARAMS=""

# OPTIONAL: user to run as
#   (format: username, default: none)
```



```
RUNASUSER="iviews"

#-----
#- script specific parameters

# pidfile location. used for stop
# (format: path+file, default: "$DIRECTORY/jobclient.pid")
PIDFILE=""

#-----
#- visualworks specific parameters

# where is the vm for that image located
# (format: path+file, default: "$DIRECTORY/vwlinux86")
VM=""
# note: on some platforms it might be necessary to prefix the vm
#       command by "setarch i386 -L " or similar, when normal starts
#       directly lead to a segmentation fault

# vm parameters ("-noherald" is usually a good idea for services)
# (format: "(-OPTION)*", default: none)
VM_PARAMS="-noherald"

#=====
#== no need to change anything below

PRODUCT="jobclient"

#-----
#- parameter checking and defaulting

if test -z "$DIRECTORY"; then DIRECTORY='.'; fi
if test ! -d "$DIRECTORY" -o ! -w "$DIRECTORY"; then
    echo "directory '$DIRECTORY' not accessible, abort"; exit 1
fi

if test -z "$VM"; then VM="$DIRECTORY/vwlinux86"; fi
if test ! -f "$VM" -o ! -x "$VM"; then
    echo "virtual machine '$VM' not accessible, abort"; exit 1
fi

if test -z "$IMAGE"; then IMAGE="$DIRECTORY/$PRODUCT.im"; fi
if test ! -f "$IMAGE" -o ! -r "$IMAGE"; then
    echo "image '$IMAGE' not accessible, abort"; exit 1
fi

if test -z "$PIDFILE"; then PIDFILE="$DIRECTORY/$PRODUCT.pid"; fi

if test -n "$INIFILE"; then PARAMS="$PARAMS -ini $INIFILE"; fi

#-----
#- actions
```





```
startproc () {
    if test $# -lt 0; then return; fi
    local CMD=$1
    local BGFG=$2
    local PIDFILE=$3
    if test -z "$CMD"; then return; fi

    CMD="$CMD <&- >&- 2>&-"
    if test "$BGFG" = bg; then
        CMD="$CMD & disown"
        if test -n "$PIDFILE"; then
            CMD="$CMD; echo \#! > '$PIDFILE'"
        fi
    fi
    fi

    if test -z "$RUNASUSER" -o "$RUNASUSER" = "$($ID -nu); then
        eval "$CMD"
    else
        $SUDO -u "$RUNASUSER" $SH -c "$CMD"
    fi
}

case "$1" in
    start)
        echo -n "Starting k-infinity $PRODUCT in $DIRECTORY: "
        cwd=$(pwd); cd $DIRECTORY
        startproc "'$VM' $VM_PARAMS '$IMAGE' $PARAMS" bg "$PIDFILE"
        cd "$cwd"
        echo "done"
        ;;
    stop)
        echo -n "Stopping k-infinity $PRODUCT in $DIRECTORY: "
        if test ! -f "$PIDFILE" -o ! -r "$PIDFILE";
        then echo "missing pidfile '$PIDFILE', abort"; exit 1
        else
            startproc "kill -KILL \$(< '$PIDFILE')" fg
            startproc "$RM -f '$PIDFILE'" fg
        fi
        echo "done"
        ;;
    restart)
        $0 stop; $0 start;
        ;;
    *)
        echo "Usage: $0 {start|stop|restart}"
        exit 1
        ;;
esac

#=====
#== the end

exit 0
```



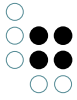
## 5.2 docker-compose Konfiguration

[.env](#)

```
MEDIATOR_PASSWORD=<mediator-password>
AUTH_TOKEN=<system-account>
```

[docker-compose.yaml](#)

```
version: '3'
services:
  mediator:
    image: container-registry.example.com/iviews-mediator:<image-tag-1>
    environment:
      IV_PASSWORD: "${MEDIATOR_PASSWORD:?MEDIATOR_PASSWORD not configured}"
      IV_SCHEDULED_JOBS: "job-bu,job-gc"
      IV_JOB_BU_VOLUME_PATTERN: "<knowledge-graph-name>"
      IV_JOB_BU_BACKUP_INTERVAL: "1"
      IV_JOB_BU_BACKUP_TIME: "22:22"
      IV_JOB_BU_BACKUPS_TO_KEEP: "5"
      IV_JOB_GC_VOLUME_PATTERN: "<knowledge-graph-name>"
      IV_JOB_GC_GARBAGE_COLLECT_INTERVAL: "1"
      IV_JOB_GC_GARBAGE_COLLECT_TIME: "22:33"
    volumes:
      - "mediator-volumes:/mediator/volumes"
      - "mediator-backup:/mediator/backup"
    ports:
      - "30000:30000"
      - "30001:30001"
    restart: unless-stopped
  jobclient:
    image: container-registry.example.com/iviews-jobclient:<image-tag-2>
    environment:
      IV_HOST: "mediator:30001"
      IV_VOLUME: "<knowledge-graph-name>"
      IV_AUTHENTICATION: "${AUTH_TOKEN:?AUTH_TOKEN not configured}"
      IV_JOB_POOLS: "script,index"
    depends_on:
      - mediator
    deploy:
      replicas: 1
    restart: unless-stopped
  bridge:
    image: container-registry.example.com/iviews-bridge:<image-tag-3>
    environment:
      IV_HOST: "mediator:30001"
      IV_VOLUME: "<knowledge-graph-name>"
      IV_AUTHENTICATION: "${AUTH_TOKEN:?AUTH_TOKEN not configured}"
```



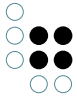
```
    depends_on:
      - mediator
      - jobclient
    ports:
      - "8815:8815"
    deploy:
      replicas: 1
      restart: unless-stopped
  volumes:
    mediator-volumes:
    mediator-backup:
```

### 5.3 kubernetes Konfiguration

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mediator
  labels:
    app: mediator
spec:
  selector:
    matchLabels:
      app: mediator
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: mediator
    spec:
      containers:
        - name: mediator
          image: container-registry.example.com/iviews-mediator:<image-tag-1>
          ports:
            - containerPort: 30000
            - containerPort: 30001
          env:
            - name: IV_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: iviews-secret
                  key: MEDIATOR_PASSWORD
            - name: IV_INTERFACES
              value: "http://0.0.0.0:30000,cnp://0.0.0.0:30001"
            - name: IV_SCHEDULED_JOBS
              value: "job-bu,job-gc"
            - name: IV_JOB_BU_VOLUME_PATTERN
              valueFrom:
                configMapKeyRef:
                  name: iviews-config
```



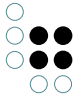
```
        key: VOLUME
      - name: IV_JOB_BU_BACKUP_INTERVAL
        value: "1"
      - name: IV_JOB_BU_BACKUP_TIME
        value: "22:22"
      - name: IV_JOB_BU_BACKUPS_TO_KEEP
        value: "1"
      - name: IV_JOB_GC_VOLUME_PATTERN
        valueFrom:
          configMapKeyRef:
            name: iviews-config
            key: VOLUME
      - name: IV_JOB_GC_GARBAGE_COLLECT_INTERVAL
        value: "1"
      - name: IV_JOB_GC_GARBAGE_COLLECT_TIME
        value: "22:33"
    volumeMounts:
      - mountPath: /mediator/volumes
        name: mediator-volumes
      - mountPath: /mediator/backup
        name: mediator-backup
    volumes:
      - name: mediator-volumes
        persistentVolumeClaim:
          claimName: mediator-volumes
      - name: mediator-backup
        persistentVolumeClaim:
          claimName: mediator-backup
    securityContext:
      runAsUser: 10000
      runAsGroup: 10000
      fsGroup: 10000
  --
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mediator-volumes
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 20Gi
  --
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mediator-backup
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
```



```
        storage: 100Gi
--
apiVersion: v1
kind: Service
metadata:
  name: mediator
  labels:
    app: mediator
spec:
  selector:
    app: mediator
  ports:
    - name: mediator-http
      port: 30000
    - name: mediator-cnp
      port: 30001
--
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: mediator
spec:
  policyTypes:
    - Ingress
  podSelector:
    matchLabels:
      app: mediator
  ingress:
    - from:
        - podSelector:
            matchLabels:
              app: jobclient
        - podSelector:
            matchLabels:
              app: bridge
--
apiVersion: apps/v1
kind: Deployment
metadata:
  name: jobclient
  labels:
    app: jobclient
spec:
  selector:
    matchLabels:
      app: jobclient
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: jobclient
    spec:
```



```
containers:
- name: jobclient
  image: container-registry.example.com/iviews-mediator:<image-tag-2>
  imagePullPolicy: IfNotPresent
  env:
  - name: IV_HOST
    value: "mediator:30001"
  - name: IV_VOLUME
    valueFrom:
      configMapKeyRef:
        name: iviews-config
        key: VOLUME
  - name: IV_AUTHENTICATION
    valueFrom:
      secretKeyRef:
        name: iviews-secret
        key: AUTHENTICATION
  - name: IV_JOB_POOLS
    value: script,index,query
securityContext:
  runAsUser: 10000
  runAsGroup: 10000
--
apiVersion: apps/v1
kind: Deployment
metadata:
  name: bridge
  labels:
    app: bridge
spec:
  selector:
    matchLabels:
      app: bridge
  strategy:
    type: Recreate
  replicas: 1
  template:
    metadata:
      labels:
        app: bridge
    spec:
      containers:
      - name: bridge
        image: container-registry.example.com/iviews-bridge:<image-tag-3>
        imagePullPolicy: IfNotPresent
        ports:
        - containerPort: 8815
        env:
        - name: IV_HOST
          value: "mediator:30001"
        - name: IV_VOLUME
          valueFrom:
            configMapKeyRef:
```



```
        name: iviews-config
        key: VOLUME
      - name: IV_AUTHENTICATION
        valueFrom:
          secretKeyRef:
            name: iviews-secret
            key: AUTHENTICATION
    securityContext:
      runAsUser: 10000
      runAsGroup: 10000
  --
apiVersion: v1
kind: Service
metadata:
  name: bridge
  labels:
    app: bridge
spec:
  type: ClusterIP
  selector:
    app: bridge
  ports:
    - name: bridge
      port: 8815
  --
apiVersion: v1
kind: Secret
metadata:
  name: iviews-secret
type: Opaque
data:
  MEDIATOR_PASSWORD: <encoded-mediator-password>
  AUTHENTICATION: <encoded-system-account>
  --
apiVersion: v1
kind: ConfigMap
metadata:
  name: iviews-config
data:
  VOLUME: <knowledge-graph-name>
```