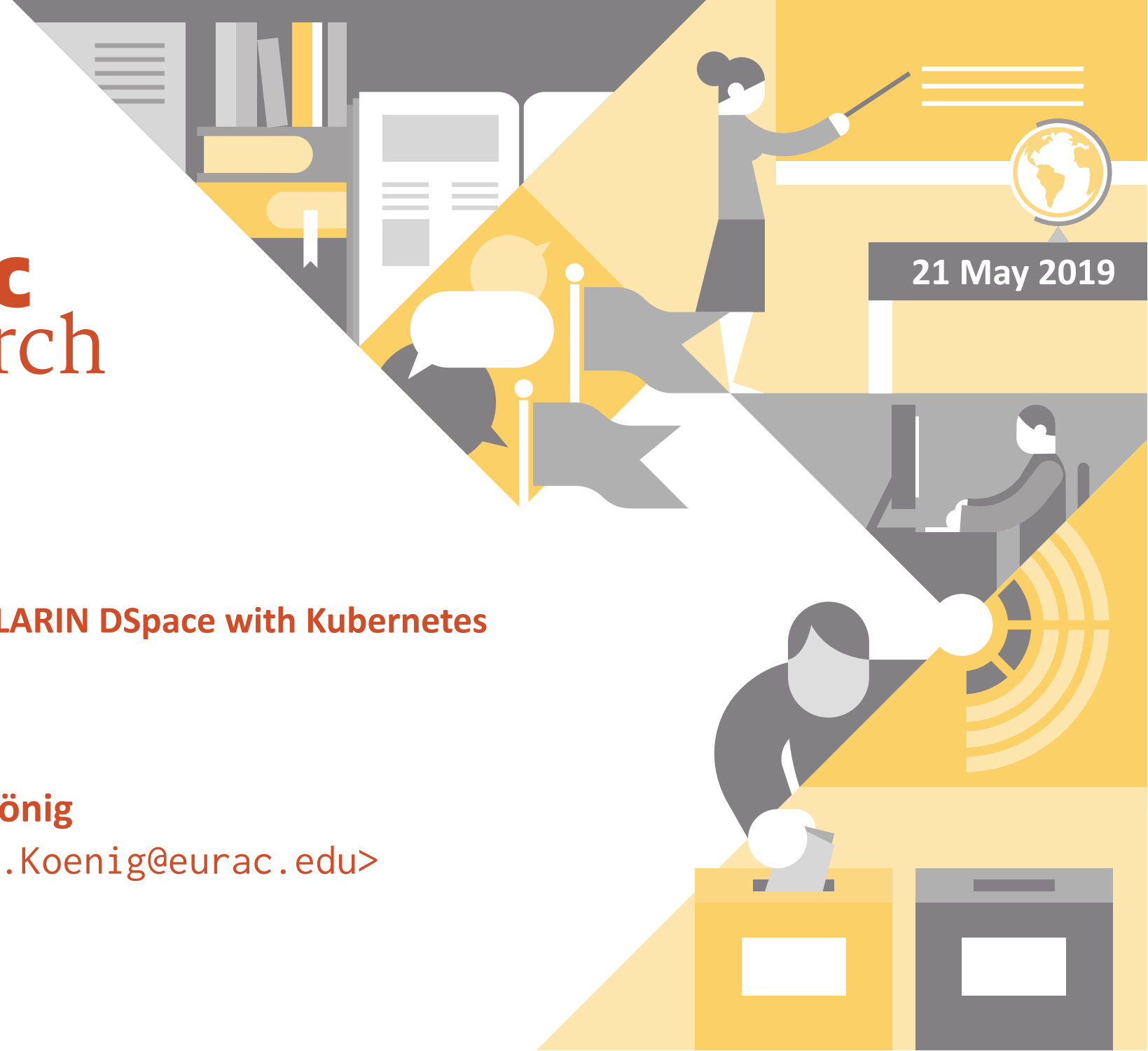

eurac research

Dockerized CLARIN DSpace with Kubernetes

Alexander König
<Alexander.Koenig@eurac.edu>

Outline

Introduction

- This presentation will give some insight into our experience with

Introduction

- This presentation will give some insight into our experience with
  - CLARIN DSpace

Introduction

- This presentation will give some insight into our experience with
  - CLARIN DSpace
  - Docker

Introduction

- This presentation will give some insight into our experience with
  - CLARIN DSpace
  - Docker
  - Kubernetes

Introduction

- This presentation will give some insight into our experience with
  - CLARIN DSpace
  - Docker
  - Kubernetes
- and how to make something new with all of them

Introduction

- This presentation will give some insight into our experience with
  - CLARIN DSpace
  - Docker
  - Kubernetes
- and how to make something new with all of them
- I'll briefly describe how we did it, where we encountered problems and how we solved them

Introduction: CLARIN DSpace

CLARIN DSpace in brief

- A fork of DSpace with some modifications and add-ons for CLARIN

Introduction: CLARIN DSpace

CLARIN DSpace in brief

- A fork of DSpace with some modifications and add-ons for CLARIN
- Developed at UFAL in Prague, but now a community project

Introduction: CLARIN DSpace

CLARIN DSpace in brief

- A fork of DSpace with some modifications and add-ons for CLARIN
- Developed at UFAL in Prague, but now a community project
- Easy and quick to set up (if you're happy with the default behavior)

Introduction: Docker

Docker in brief

- Under heavy development (sometimes still breaks backwards compatibility)

Introduction: Docker

Docker in brief

- Under heavy development (sometimes still breaks backwards compatibility)
- Available for all major OSes (with a focus on Unix-based ones)

Introduction: Docker

Docker in brief

- Under heavy development (sometimes still breaks backwards compatibility)
- Available for all major OSes (with a focus on Unix-based ones)
- The most popular containerization software

Introduction: Docker

Docker in brief

- Under heavy development (sometimes still breaks backwards compatibility)
- Available for all major OSes (with a focus on Unix-based ones)
- The most popular containerization software
  - Bundle all necessary software and libraries

Introduction: Docker

Docker in brief

- Under heavy development (sometimes still breaks backwards compatibility)
- Available for all major OSes (with a focus on Unix-based ones)
- The most popular containerization software
  - Bundle all necessary software and libraries
  - Without all the bloat that is not needed (but installed by default in a VM)

Introduction: Docker

Docker in brief

- Under heavy development (sometimes still breaks backwards compatibility)
- Available for all major OSes (with a focus on Unix-based ones)
- The most popular containerization software
  - Bundle all necessary software and libraries
  - Without all the bloat that is not needed (but installed by default in a VM)
  - Closer coupling to the host resources (less overhead)

Introduction: Kubernetes

Kubernetes in brief

- Orchestration tool for (Docker) containers

Introduction: Kubernetes

Kubernetes in brief

- Orchestration tool for (Docker) containers
- Helps with

Introduction: Kubernetes

Kubernetes in brief

- Orchestration tool for (Docker) containers
- Helps with
  - Managing volumes (e.g. through connecting a ceph filesystem)

Introduction: Kubernetes

Kubernetes in brief

- Orchestration tool for (Docker) containers
- Helps with
  - Managing volumes (e.g. through connecting a ceph filesystem)
  - Scaling by spinning up more containers and

Introduction: Kubernetes

Kubernetes in brief

- Orchestration tool for (Docker) containers
- Helps with
  - Managing volumes (e.g. through connecting a ceph filesystem)
  - Scaling by spinning up more containers and
  - replacing dead containers by restarting them

Introduction: Kubernetes

Kubernetes in brief

- Orchestration tool for (Docker) containers
- Helps with
  - Managing volumes (e.g. through connecting a ceph filesystem)
  - Scaling by spinning up more containers and
  - replacing dead containers by restarting them
  - Managing (HTTP and HTTPS) access from outside through a central nginx proxy (Ingress)

Outline

The Idea

The situation

- IAL at Eurac Research wanted to set up a repository to become a CLARIN Centre

The Idea

The situation

- IAL at Eurac Research wanted to set up a repository to become a CLARIN Centre
- CLARIN DSpace quickly turned out to be the obvious choice

The Idea

The situation

- IAL at Eurac Research wanted to set up a repository to become a CLARIN Centre
- CLARIN DSpace quickly turned out to be the obvious choice
- We decided to dockerize it for multiple reasons

The Idea

The situation

- IAL at Eurac Research wanted to set up a repository to become a CLARIN Centre
- CLARIN DSpace quickly turned out to be the obvious choice
- We decided to dockerize it for multiple reasons
  - it seems like a cleaner, more easily reproducible set-up than doing it inside a VM

The Idea

The situation

- IAL at Eurac Research wanted to set up a repository to become a CLARIN Centre
- CLARIN DSpace quickly turned out to be the obvious choice
- We decided to dockerize it for multiple reasons
  - it seems like a cleaner, more easily reproducible set-up than doing it inside a VM
  - Our IT had recently set up a Kubernetes cluster

The Idea

The situation

- IAL at Eurac Research wanted to set up a repository to become a CLARIN Centre
- CLARIN DSpace quickly turned out to be the obvious choice
- We decided to dockerize it for multiple reasons
  - it seems like a cleaner, more easily reproducible set-up than doing it inside a VM
  - Our IT had recently set up a Kubernetes cluster
  - CLARIN-IT is still in the beginning and doesn't have much money, so we thought creating a (more or less) one-click installation for a CLARIN-compatible repo might be interesting to other institutes in Italy

Outline

The Implementation: All-in-One

One Container To Rule Them All

- As a first step (as Docker newbies) we followed the DSpace installation instructions and set it all up within one container

The Implementation: All-in-One

One Container To Rule Them All

- As a first step (as Docker newbies) we followed the DSpace installation instructions and set it all up within one container
- Typical docker pitfalls
  - some system services like systemd are not available
  - some helper programs (less,vim,ping) are not available (for debugging)
  - communications to the outside can be tricky (especially non HTPP(S))

The Implementation: All-in-One

One Container To Rule Them All

- As a first step (as Docker newbies) we followed the DSpace installation instructions and set it all up within one container
- Typical docker pitfalls
  - some system services like systemd are not available
  - some helper programs (less,vim,ping) are not available (for debugging)
  - communications to the outside can be tricky (especially non HTPP(S))
- The installation process needed some manual steps (e.g. generating the admin user) which we needed to automatize

The Implementation: All-in-One

One Container To Rule Them All

- As a first step (as Docker newbies) we followed the DSpace installation instructions and set it all up within one container
- Typical docker pitfalls
  - some system services like systemd are not available
  - some helper programs (less,vim,ping) are not available (for debugging)
  - communications to the outside can be tricky (especially non HTPP(S))
- The installation process needed some manual steps (e.g. generating the admin user) which we needed to automatize
- End result: ”Dockerized CLARIN DSpace”

The Implementation: Docker Compose

Separating Services using Docker Compose

- We now have three containers nginx including shibboleth, psql and a big one with all the rest

The Implementation: Docker Compose

Separating Services using Docker Compose

- We now have three containers nginx including shibboleth, psql and a big one with all the rest
- Separation problems

The Implementation: Docker Compose

Separating Services using Docker Compose

- We now have three containers nginx including shibboleth, psql and a big one with all the rest
- Separation problems
  - Most services talk to each other over the network, but some need local files or sockets

The Implementation: Docker Compose

Separating Services using Docker Compose

- We now have three containers nginx including shibboleth, psql and a big one with all the rest
- Separation problems
  - Most services talk to each other over the network, but some need local files or sockets
  - Figuring out which libraries are necessary for which program is not always easy

The Implementation: Docker Compose

Separating Services using Docker Compose

- We now have three containers nginx including shibboleth, psql and a big one with all the rest
- Separation problems
  - Most services talk to each other over the network, but some need local files or sockets
  - Figuring out which libraries are necessary for which program is not always easy
- It became obvious that some internal program manager (e.g. supervisor) is often necessary

The Implementation: Kubernetes

Konverting Docker Compose to Kubernetes

- Handy tool Kompose (`https://kompose.io/`) creates basic Kubernetes yaml files from docker-compose.yml

The Implementation: Kubernetes

Konverting Docker Compose to Kubernetes

- Handy tool Kompose (`https://kompose.io/`) creates basic Kubernetes yaml files from docker-compose.yml
- Volumes need to be added using Kubernetes volume management (we use a ceph filesystem that is connected to the Kube cluster)

The Implementation: Kubernetes

Konverting Docker Compose to Kubernetes

- Handy tool Kompose (`https://kompose.io/`) creates basic Kubernetes yaml files from docker-compose.yml
- Volumes need to be added using Kubernetes volume management (we use a ceph filesystem that is connected to the Kube cluster)
- All secret information (e.g. usernames and passwords) is moved to Kubernetes secrets

The Implementation: Kubernetes

Konverting Docker Compose to Kubernetes

- Handy tool Kompose (`https://kompose.io/`) creates basic Kubernetes yaml files from docker-compose.yml
- Volumes need to be added using Kubernetes volume management (we use a ceph filesystem that is connected to the Kube cluster)
- All secret information (e.g. usernames and passwords) is moved to Kubernetes secrets
- Every outside access to containers is managed by a central Ingress proxy where also the SSL certificates are stored

The Implementation: Kubernetes

Konverting Docker Compose to Kubernetes

- Handy tool Kompose (`https://kompose.io/`) creates basic Kubernetes yaml files from docker-compose.yml
- Volumes need to be added using Kubernetes volume management (we use a ceph filesystem that is connected to the Kube cluster)
- All secret information (e.g. usernames and passwords) is moved to Kubernetes secrets
- Every outside access to containers is managed by a central Ingress proxy where also the SSL certificates are stored
  - makes it hard to distinguish different services via IP (setting up two handle servers)

The Implementation: Kubernetes

Konverting Docker Compose to Kubernetes

- Handy tool Kompose (`https://kompose.io/`) creates basic Kubernetes yaml files from docker-compose.yml
- Volumes need to be added using Kubernetes volume management (we use a ceph filesystem that is connected to the Kube cluster)
- All secret information (e.g. usernames and passwords) is moved to Kubernetes secrets
- Every outside access to containers is managed by a central Ingress proxy where also the SSL certificates are stored
  - makes it hard to distinguish different services via IP (setting up two handle servers)
  - non-standard ports (e.g. handle net) are difficult to set up

The Implementation: Kubernetes

Konverting Docker Compose to Kubernetes

- Handy tool Kompose (`https://kompose.io/`) creates basic Kubernetes yaml files from docker-compose.yml
- Volumes need to be added using Kubernetes volume management (we use a ceph filesystem that is connected to the Kube cluster)
- All secret information (e.g. usernames and passwords) is moved to Kubernetes secrets
- Every outside access to containers is managed by a central Ingress proxy where also the SSL certificates are stored
  - makes it hard to distinguish different services via IP (setting up two handle servers)
  - non-standard ports (e.g. handle net) are difficult to set up
- Kubernetes fetches Docker images from registry (but as there are multiple nodes, sometimes the same image needs to be reloaded upon restart, which is mostly a problem with large images)

The Implementation: Customization

Implementing Eurac Branding and Adapting some defaults

- Used CLARIN WAYF instead of built-in UFAL one

The Implementation: Customization

Implementing Eurac Branding and Adapting some defaults

- Used CLARIN WAYF instead of built-in UFAL one
- Adapted documentation and privacy policy (GDPR)

The Implementation: Customization

Implementing Eurac Branding and Adapting some defaults

- Used CLARIN WAYF instead of built-in UFAL one
- Adapted documentation and privacy policy (GDPR)
- Adapted color style to Eurac (proved problematic because some functionality seems to be hardcoded in the UFAL theme)

The Implementation: Customization

Implementing Eurac Branding and Adapting some defaults

- Used CLARIN WAYF instead of built-in UFAL one
- Adapted documentation and privacy policy (GDPR)
- Adapted color style to Eurac (proved problematic because some functionality seems to be hardcoded in the UFAL theme)
- Added a static start page that introduces the ERCC

Outline

The Problems: Versioning of software libraries

Software used in the Dockerfile has to be strictly versioned

- Our DSpace Dockerfile uses a lot of sometimes very specific software

The Problems: Versioning of software libraries

Software used in the Dockerfile has to be strictly versioned

- Our DSpace Dockerfile uses a lot of sometimes very specific software
- Some like Shibboleth are built from various sources that are fetched with a script

The Problems: Versioning of software libraries

Software used in the Dockerfile has to be strictly versioned

- Our DSpace Dockerfile uses a lot of sometimes very specific software
- Some like Shibboleth are built from various sources that are fetched with a script
- For some libraries the URL where the source can be downloaded cannot be guessed from the version number alone (manual investigation necessary)

The Problems: Versioning of software libraries

Software used in the Dockerfile has to be strictly versioned

- Our DSpace Dockerfile uses a lot of sometimes very specific software
- Some like Shibboleth are built from various sources that are fetched with a script
- For some libraries the URL where the source can be downloaded cannot be guessed from the version number alone (manual investigation necessary)
- Sometimes working URLs stop resolving because old library versions are archived in some way (=> build script suddenly stops working)

The Problems: General Kubernetes Problems

Kubernetes Problems and Peculiarities

- Proxying every access through Ingress needed some config tweaking to not lose origin IP (for statistics)

The Problems: General Kubernetes Problems

Kubernetes Problems and Peculiarities

- Proxying every access through Ingress needed some config tweaking to not lose origin IP (for statistics)
- Kubernetes often assumes that every user is admin which means that some info is not available to a non-privileged user (e.g. Ingress logs => makes debugging harder)

The Problems: General Kubernetes Problems

Kubernetes Problems and Peculiarities

- Proxying every access through Ingress needed some config tweaking to not lose origin IP (for statistics)
- Kubernetes often assumes that every user is admin which means that some info is not available to a non-privileged user (e.g. Ingress logs => makes debugging harder)
- Kubernetes is heavily developed, which means that sometimes urgent updates are needed, which may result in some services becoming unavailable

The Problems: Various Issues

Open Issues and Common Problems

- Implementing CLARIN WAYF resulted in local login to be unavailable

The Problems: Various Issues

Open Issues and Common Problems

- Implementing CLARIN WAYF resulted in local login to be unavailable
- Rebasing lindat-common (and to a lesser degree) clarin-dspace repo often results in conflicts because we had to change some general things

The Problems: Various Issues

Open Issues and Common Problems

- Implementing CLARIN WAYF resulted in local login to be unavailable
- Rebasing lindat-common (and to a lesser degree) clarin-dspace repo often results in conflicts because we had to change some general things
- Eurac theming is spread across three projects (forks of clarin-dspace & lindat-common and the general docker project)

Outline

What comes next?

Open Issues

- Figure out a way to have both Shibboleth and local login available

What comes next?

Open Issues

- Figure out a way to have both Shibboleth and local login available
- Investigate whether/how the handle server can be moved to its own container

What comes next?

Open Issues

- Figure out a way to have both Shibboleth and local login available
- Investigate whether/how the handle server can be moved to its own container
- Improve documentation

What comes next?

Open Issues

- Figure out a way to have both Shibboleth and local login available
- Investigate whether/how the handle server can be moved to its own container
- Improve documentation
- (If possible) enable docker-compose again (in a separate branch?) for users without Kubernetes

What comes next?

Open Issues

- Figure out a way to have both Shibboleth and local login available
- Investigate whether/how the handle server can be moved to its own container
- Improve documentation
- (If possible) enable docker-compose again (in a separate branch?) for users without Kubernetes
- Shibbolize our search interfaces (e.g. Annis) and integrate them with DSpace

What comes next?

Open Issues

- Figure out a way to have both Shibboleth and local login available
- Investigate whether/how the handle server can be moved to its own container
- Improve documentation
- (If possible) enable docker-compose again (in a separate branch?) for users without Kubernetes
- Shibbolize our search interfaces (e.g. Annis) and integrate them with DSpace
- Investigate the use of a templating engine (e.g. Helm) to make updates more robust (e.g. version number only needs to be changed once)

What comes next?

Open Issues

- Figure out a way to have both Shibboleth and local login available
- Investigate whether/how the handle server can be moved to its own container
- Improve documentation
- (If possible) enable docker-compose again (in a separate branch?) for users without Kubernetes
- Shibbolize our search interfaces (e.g. Annis) and integrate them with DSpace
- Investigate the use of a templating engine (e.g. Helm) to make updates more robust (e.g. version number only needs to be changed once)
- Various optimizations (e.g. reduce the container size, main container is currently > 3GB)

Fin

Thank you for your attention!

https://gitlab.inf.unibz.it/commul/docker/clarin-dspace/

Alexander.Koenig@eurac.edu