

HEY, HELM, CAN YOU SCALE?

@jbaruch #VelocityConf @HelmPack jfrog.com/shownotes

ABOUT ME

THAT'S WHAT I DO:
I DRINK AND
I KNOW THINGS.



@jbaruch



SHOWNOTES

- <https://jfrog.com/shownotes>
 - The slides
 - The video (by tomorrow)
 - All the links
 - Comments, ratings
 - Raffle!

POLL TIME!

Let's calibrate the level first

@jbaruch #VelocityConf @HelmPack jfrog.com/shownotes

HELM WHAT?

- Heard about Helm
- Played a bit with Helm
- Using Helm in non-prod environments
- Using Helm in production
- I wrote this thing

WHAT THE HELM IS HELM?

Dependency manager for Kubernetes

HOW TO DEPLOY ANYTHING TO K8S

- Copy YAML
- Paste YAML
- Fix indents
- Repeat

KUBERNETES RESOURCE

```
{  
  "kind": "Deployment",  
  "apiVersion": "extensions/v1beta1",  
  "metadata": {  
    "name": "my-release-docker-app-chart"  
  },  
  "spec": {  
    "containers": [  
      {  
        "name": "docker-app-chart",  
        "image": "docker.artifactory/docker-app:1.0",
```


LET'S BUILD A NEW ONE!

```
> docker build -t docker.artifactory/docker-app:1.1
```

ONE LAST THING...

```
> sed -i.bak  
`s#docker.artifactory/docker-app:1.1#{imageTag}#`  
deployment.yaml
```



@jbaruch #VelocityConf @HelmPack jfrog.com/shownotes

OR JUST USE :LATEST

```
"image": "docker.artifactory/docker-app:latest"
```



ENTER HELM

@jbaruch #VelocityConf @HelmPack jfrog.com/shownotes

ENCAPSULATED PACKAGES OF KUBERNETES DEPLOYMENTS

All this...

xrayxray-analysis

xray-event

xray-indexer

xray-nfs-server

xray-persist

Becomes this

xray

POWERFUL TEMPLATING FOR DESCRIPTOR FILES

```
{
  "kind": "Deployment",
  "apiVersion": "extensions/v1beta1",
  "metadata": {
    "name": "{{ template \"docker-app.fullname\" . }}"
  },
  "spec": {
    "containers": [
      {
        "name": "{{ template \"docker-app.name\" . }}",
        "image": "{{ .Values.image.repository }}:
                  {{ .Values.image.tag }}"
      }
    ]
  }
}
```

VALUES:

```
# Default values for docker-app.  
# This is a YAML-formatted file.  
# Declare name/value pairs to be passed into your templates.
```

image:

repository: docker.artifactory/docker-app

tag: 1.1

secretName: regsecret

pullPolicy: Always

SIMPLE!

- Templates
- Values
- Metadata

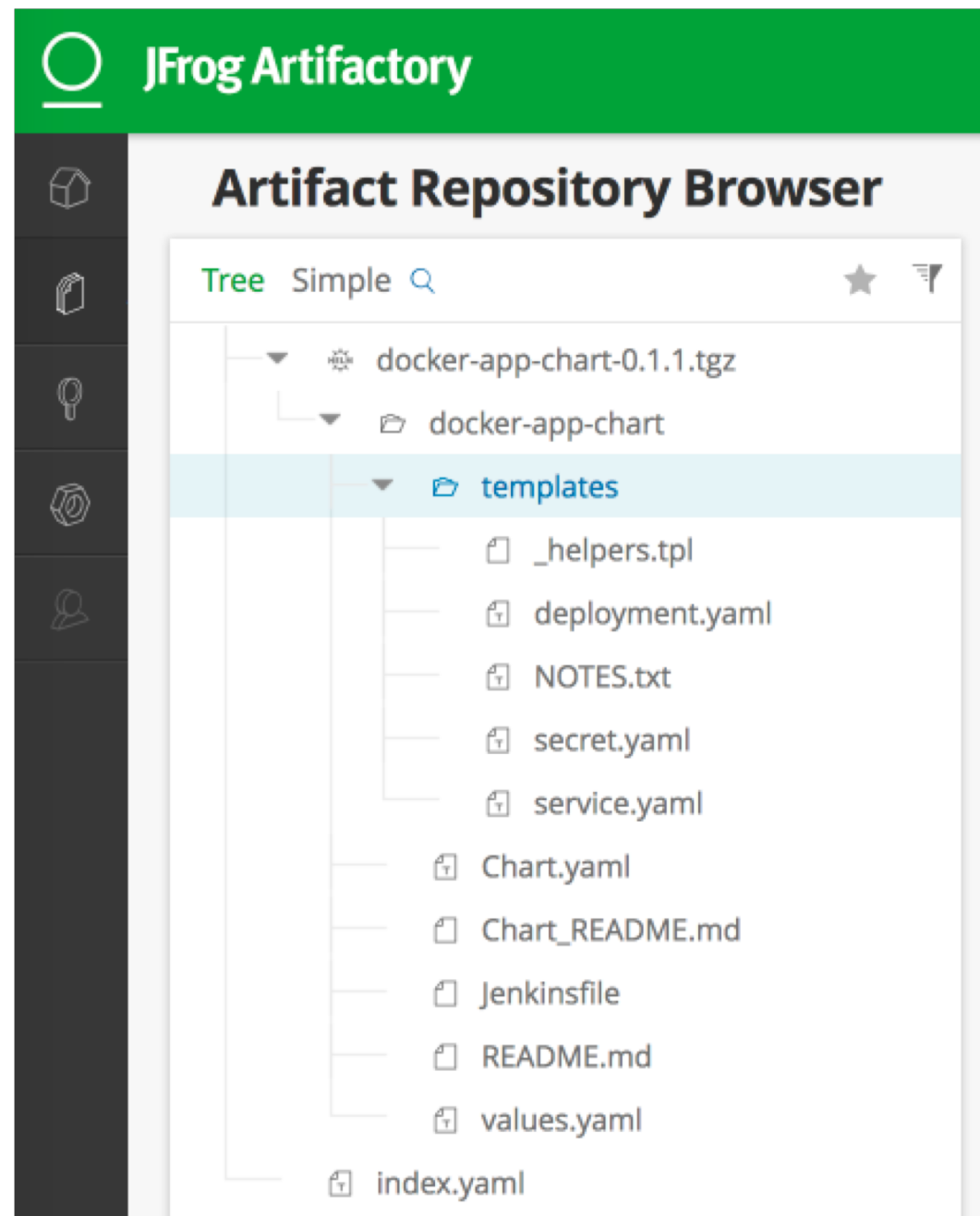
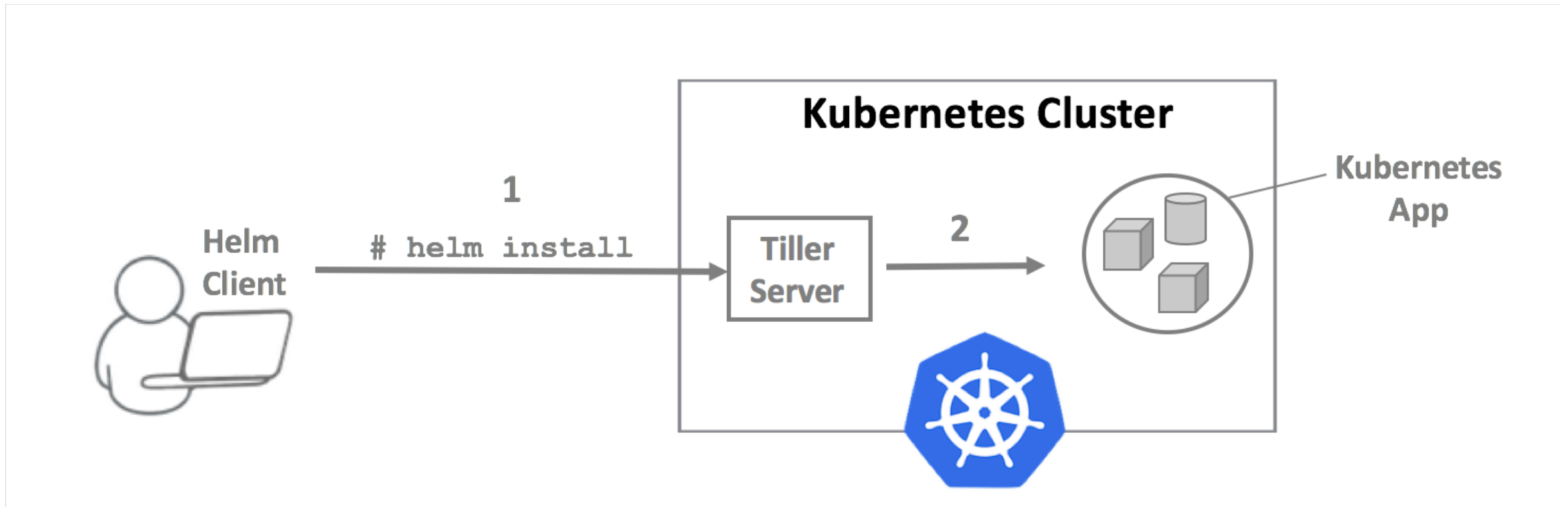


CHART <-> IMAGE RELATIONSHIP

- Using templates we can reuse charts for multiple image tags
- Chart versions != Image tags
- Or are they...?



KUBERNETES CLUSTER CONTROL



TWO PARTS

Helm client

- Local chart development
- Managing repositories
- Interacting with the Tiller server

Tiller Server

- Listening for incoming requests from the Helm client
- Combining a chart and configuration to build a release
- Installing charts into Kubernetes, and then tracking the subsequent release
- Upgrading and uninstalling charts by interacting with Kubernetes

HELM COMMANDS

>helm init

>helm search

>helm install

>helm status

>helm repo


HELM REPOSITORIES

- Official repository - kubernetes.io/helm/


Discover & launch great Kubernetes-ready apps

Search charts...

159 charts ready to deploy




acs-engine-autoscaler
2.1.1 *stable* ☆ 0



aerospike
v3.14.1.2 *stable* ☆ 0




anchore-engine
0.1.6 *stable* ☆ 0




artifactory
5.8.3 *stable* ☆ 17




buildkite
3 *stable* ☆ 1



burrow
0.12.0 *Incubator* ☆ 0



cassandra
3.11.0 *Incubator* ☆ 3



centrifugo
1.7.3 *stable* ☆ 0

@jbaruch #VelocityConf @HelmPack jfrog.com/shownotes

HELM REPOSITORIES

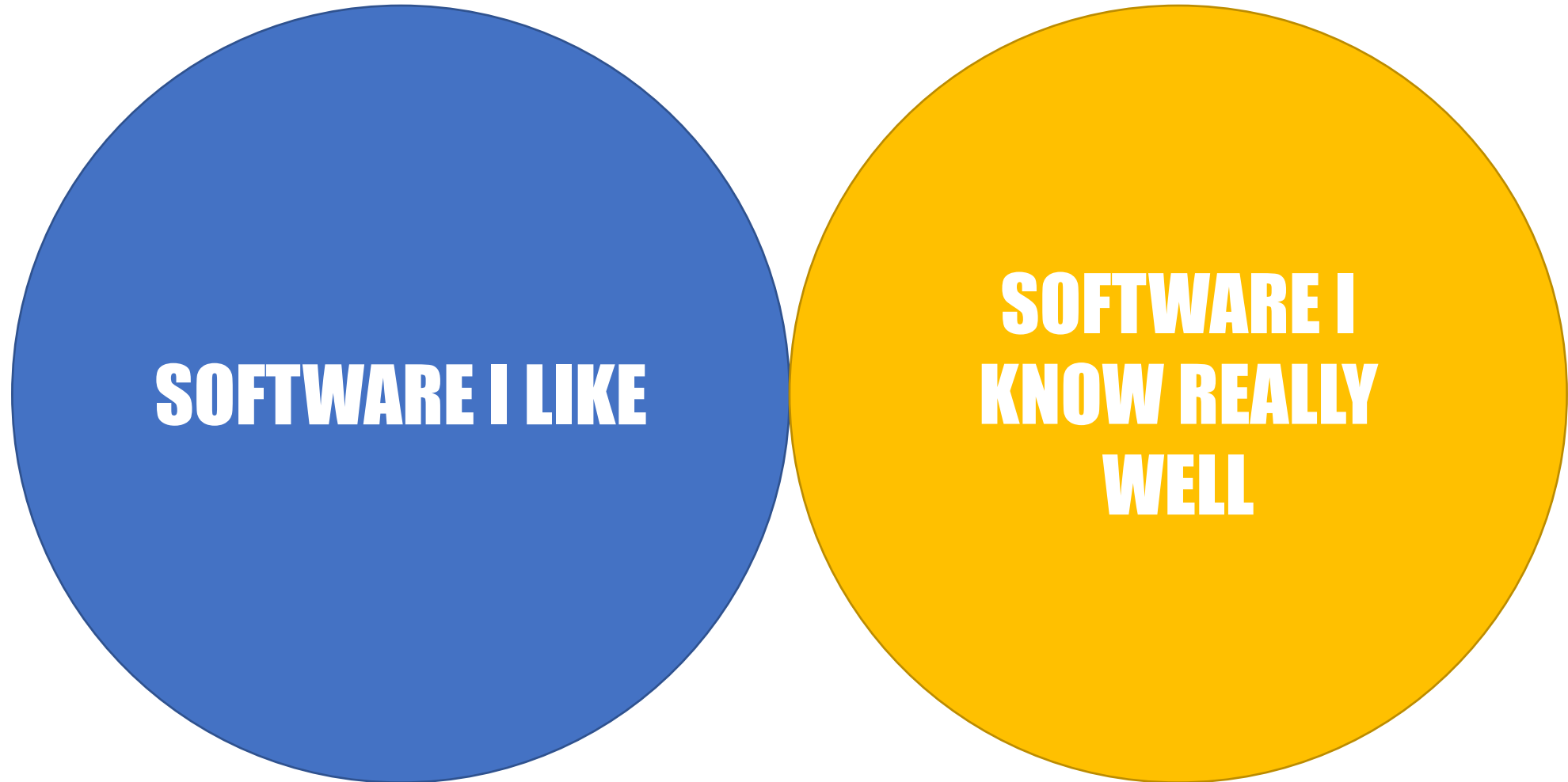
- Official repository - kubernetes.io/helm/
- Get a local one!
- Option 1: Create your own:
 - Run an http server with `index.yaml`
 - Run `helm repo index` to generate one the index
- Option 2: Use JFrog Artifactory (or others)
 - Kubernetes Registry which supports Helm, containers, and everything inside them

HELM 3

- No tiller (who needs it anyway)
- Lua instead of YAML



SOFTWARE ENGINEERING VENN DIAGRAM



WHAT DEPENDENCY MANAGERS AND PRINTERS HAVE IN COMMON?

Why I Believe

Printers



Were Sent From

Hell

To Make Us Miserable

By The Oatmeal <http://theoatmeal.com>



So you want to write a package manager

You woke up this morning, rolled out of bed, and thought, “Y’know what? I don’t have enough misery and suffering in my life. I know what to do—I’ll write a language package manager!”

...

Package management is awful, you should quit right now

Package management is a nasty domain. Really nasty. On the surface, it *seems* like a purely technical problem, amenable to purely technical solutions. And so, quite reasonably, people approach it that way. Over time, these folks move inexorably towards the conclusion that:

1. software is terrible
2. people are terrible
3. there are too many different scenarios
4. nothing will really work for sure
5. it’s provable that nothing will really work for sure
6. **our lives are meaningless perturbations in a swirling vortex of chaos and entropy**

★ Top highlight



sam boyer

@sdboyer

systems | people

📍 Detroit metro

📅 Joined December 2008

7 DEADLY SINS OF PACKAGE MANAGERS

1. Over-architecture
2. Not thinking of enterprise scenarios
3. Having downloadable index
4. Cross-site dependency resolution loopholes
5. Author authentication done wrong
6. Version management (or lack of thereof)
7. Using wrong place for central registry (and hardcoding it!)

7 DEADLY SINS OF PACKAGE MANAGERS

- ~~1. Over-architecture~~
- ~~2. Not thinking of enterprise scenarios~~
- ~~3. Having downloadable index~~
- ~~4. Cross-site dependency resolution loopholes~~
- ~~5. Author authentication done wrong~~
- ~~6. Version management (or lack of thereof)~~
- ~~7. Using wrong place for central registry (and hardcoding it!)~~

ENTERPRISE SCENARIOS

- In-house registry
- User authentication and authorization
- org/project/team segmentation with central management
- Promotion pipelines



Authentication support for remote charts repositories #3206

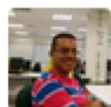
Merged bacongobbler merged 33 commits into `kubernetes:master` from `eyalbe4:master` 14 days ago

Conversation 72

Commits 33

Files changed 25

+256 -108



eyalbe4 commented on Nov 27, 2017

Contributor +

This pull request adds support for sending credentials (username and password) to remote charts repositories. This will allow vendors of charts repositories to restrict access to privileged users only. This new functionality is optional of course.

Please let us know what you think.
We'd be happy to add more commits as needed.
Thank you in advance.

👍 19

Reviewers

- bacongobbler ✓
- michelleN ✓
- seriousben ✓

Assignees

No one assigned

Labels

conf-cla: yes

DOWNLOADABLE INDEX?

- Pros:

- Simple server
- Offline search
- Offload the search



er power


DOWNLOADABLE INDEX?

- Pros:
 - Simple server
 - Offline search (not sure how it's useful?)
 - Offload the search off the server to spare computer power
- Cons:
 - Trends to become obsolete
 - Index is a bottleneck (specially if done wrong)

LET'S RUN SOME TESTS ON 120K OF CHARTS...


@jbaruch #VelocityConf @HelmPack jfrog.com/shownotes

120K OF CHARTS?!

- True, you can reuse same charts for different image tags
- What will be the value in values.yaml?
 - LATEST probably 
- Every time you run you have to specify the image tag



120K OF CHARTS?!

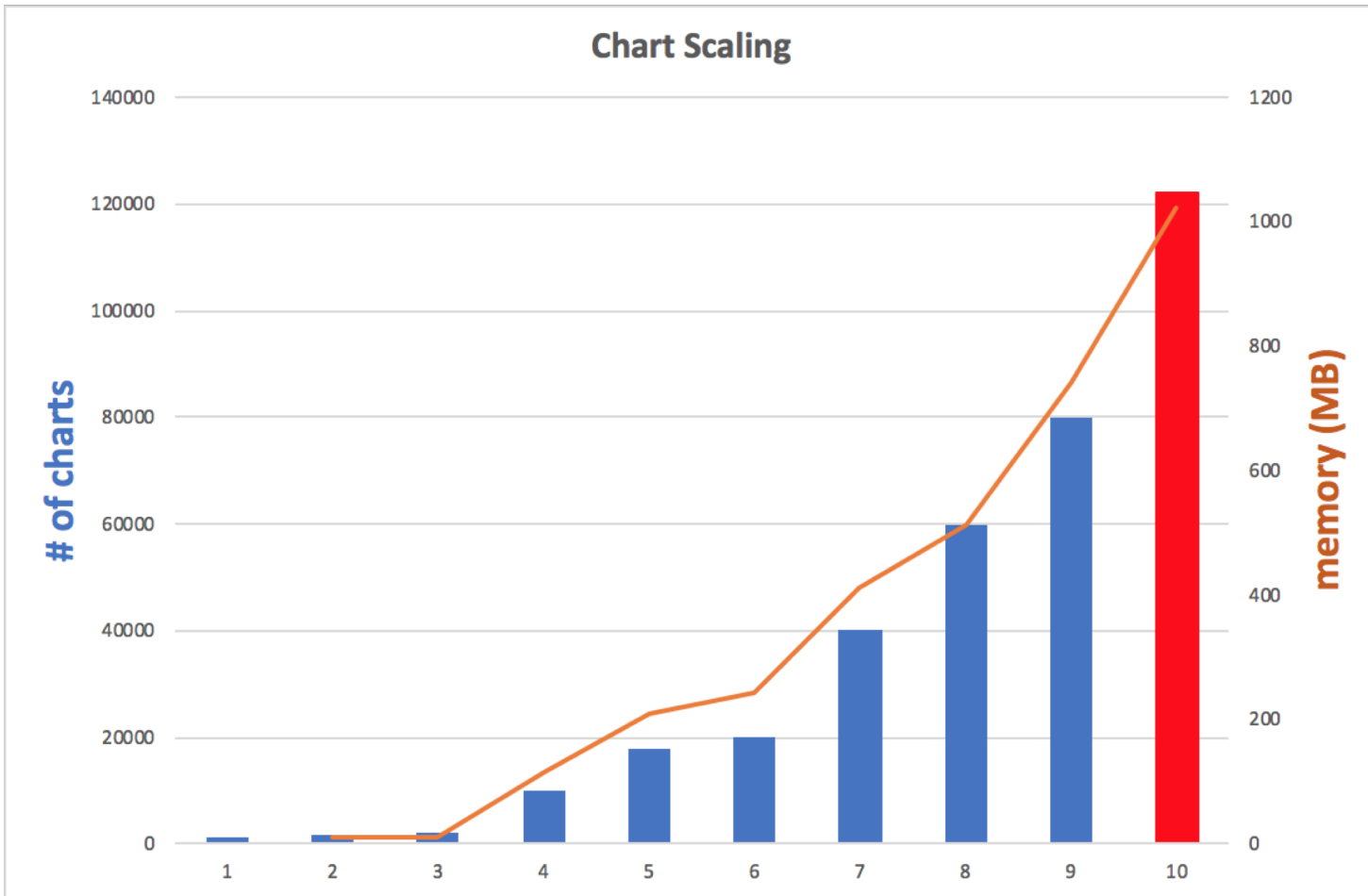
- True, you can reuse same charts for different image tags
- What will be the value in values.yaml?
 - LATEST probably 
- Every time you run you have to specify the image tag



**YOU'LL HAVE A CARTESIAN PRODUCT OF CHARTS FROM THE ALL THE
TAGS OF ALL THE IMAGES IN THE CHARTS.**

THAT'S A LOT OF CHARTS.

INDEX BOTTLENECKS



```
docker run -m=512m -it alpine-helm  
helm repo add helm-prod $helm-repo
```


HOW CAN WE FIX IT?

- gzip index in transit
 - Fixes **some** of the issues
- Distribute the index

artifactory:

← Entry per app

```
- apiVersion: v1
  appVersion: 5.8.3
  created: 2018-01-29T21:34:25.891969615Z
  description: Universal Repository Manager supporting all major packaging formats,
    build tools and CI servers.
  digest: c66011c6837d1fc27ffe261e8e0205188a398570b30cc61f32dcfcfb4f9f19a6
  home: https://www.jfrog.com/artifactory/
  icon: https://raw.githubusercontent.com/JFrogDev/artifactory-dcos/master/images/jfrog_med.png
  keywords:
  - artifactory
  - jfrog
  maintainers:
  - email: jainishs@jfrog.com
    name: jainishshah17
  - email: eldada@jfrog.com
    name: eldada
  name: artifactory
  sources:
  - https://bintray.com/jfrog/product/JFrog-Artifactory-Pro/view
  - https://github.com/JFrogDev
  urls:
  - https://kubernetes-charts.storage.googleapis.com/artifactory-6.2.5.tgz
  version: 6.2.5
```

```
- apiVersion: v1
  appVersion: 5.6.2
  created: 2017-12-18T20:03:55.777635493Z
  description: Universal Repository Manager supporting all major packaging formats,
    build tools and CI servers.
```

← Entry per version

DIVIDE AND CONQUER!

- Main index: list of apps (with latest version)
 - artifactory:5.8.3
- App index: list of versions (and app-level metadata)
 - description
 - maintainers
 - keywords
 - sources
- Version index: the details of the version
 - appVersion
 - created
 - digest
 - url

STRUCTURED INDEX REQUIRES STRUCTURED REPOSITORY

- Layout!
- Repo
 - App
 - ver1
 - ver2
- This complicates the push!
 - We used to just upload the file!
 - See debs (or don't)

LET'S TALK ABOUT PUSH

Pushing Charts to a Repository

TODO: notes on usage of `helm push`

Several registries have emerged with support for Helm charts including JFrog Artifactory, ChartMuseum and Quay (via Application Registries) along with extras built on top of other systems, such as object storage (e.g., see the [S3 Plugin](#)). These registries are asking for and could benefit from a capability to push packages from the Helm client.

To support registries there will be a `helm push` command that works in a similar manner to `helm fetch`. It will support different systems via the URI/URL scheme and provide a default implementation for HTTP(S). Other systems, such as S3, can implement an uploader via a plugin for schemes Helm does not provide an implementation for.

Q&A AND TWITTER ADS

- @jbaruch
- #VelocityConf
- @HelmPack
- <https://jfrog.com/shownotes>