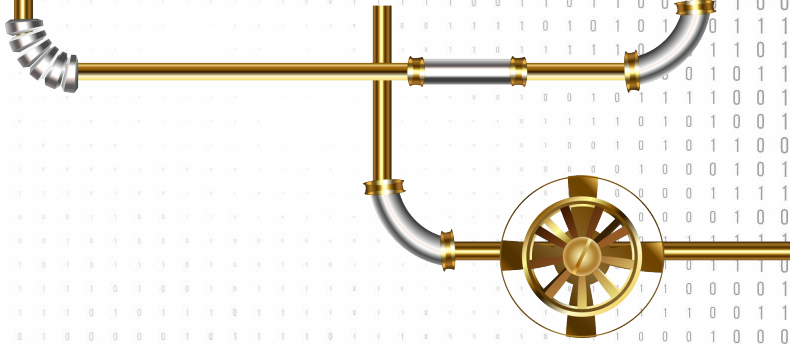


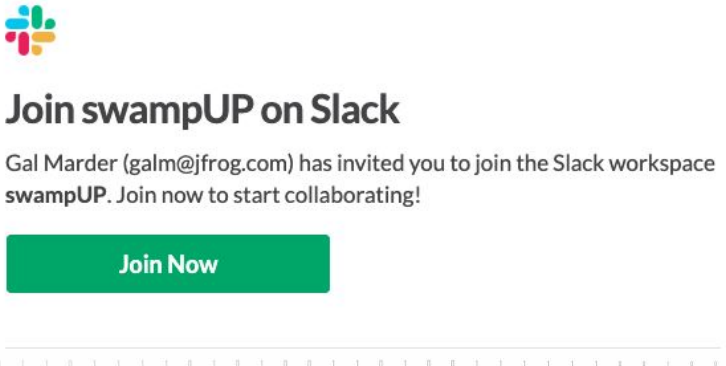
JFrog 2019 swampUP

TROUBLESHOOTING & MONITORING ARTIFACTORY

Elad Hirsch - https://www.linkedin.com/in/eladhirsch/
Devops Consultant

slack Setup for Course Collaboration

- Check your email for an invitation to join a Slack workspace
- Click on the ‘Join Now’ icon
- Verify your name
- Create a password and click ‘Create Account’
- Agree to the ‘Slack Terms of Service’
- Workspace URL: swampup.slack.com

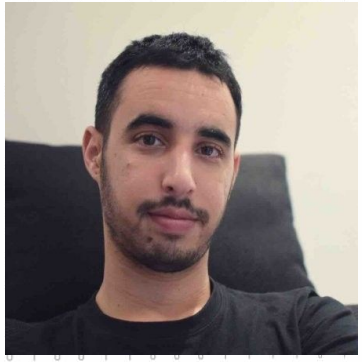


WHO'S WHO?

Elad Hirsch
DevOps Consultant

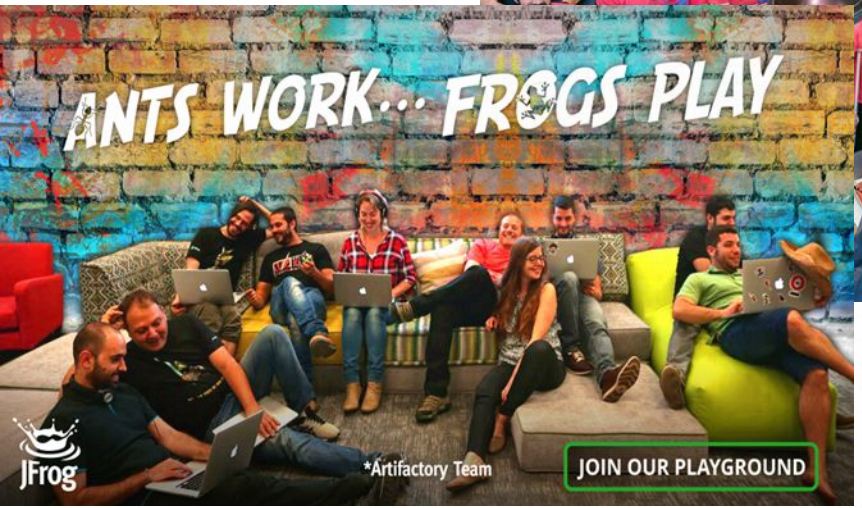
Ariel Seftel Manager
High Touch Solutions

Maxim Yurkovsky
Senior Engineer



Uriah Levy
Senior Engineer

FROGS?



JFROG IN A NUTSHELL

2008
Founded

5,000
Customers

400
Employees

FORTUNE 100 COMPANY
Clients include
70%
of the
Fortune 100

$230M
Raised to date

All Hybrid
From OSS to Multicloud

6 Products
from Git to K8S

9 locations
7 countries

TECHNOLOGY LEADERSHIP

Forbes CLOUD 100 LIST

Deloitte 2018 Technology Fast 500 Winners

The 2018 SD Times 100 Award

JFrog

THE FROG PHILOSOPHY

END-TO-END PLATFORM

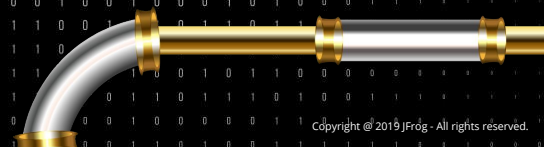
CONTINUOUS SECURITY

SCALES TO INFINITY

HYBRID AND MULTI-CLOUD

RADICALLY UNIVERSAL

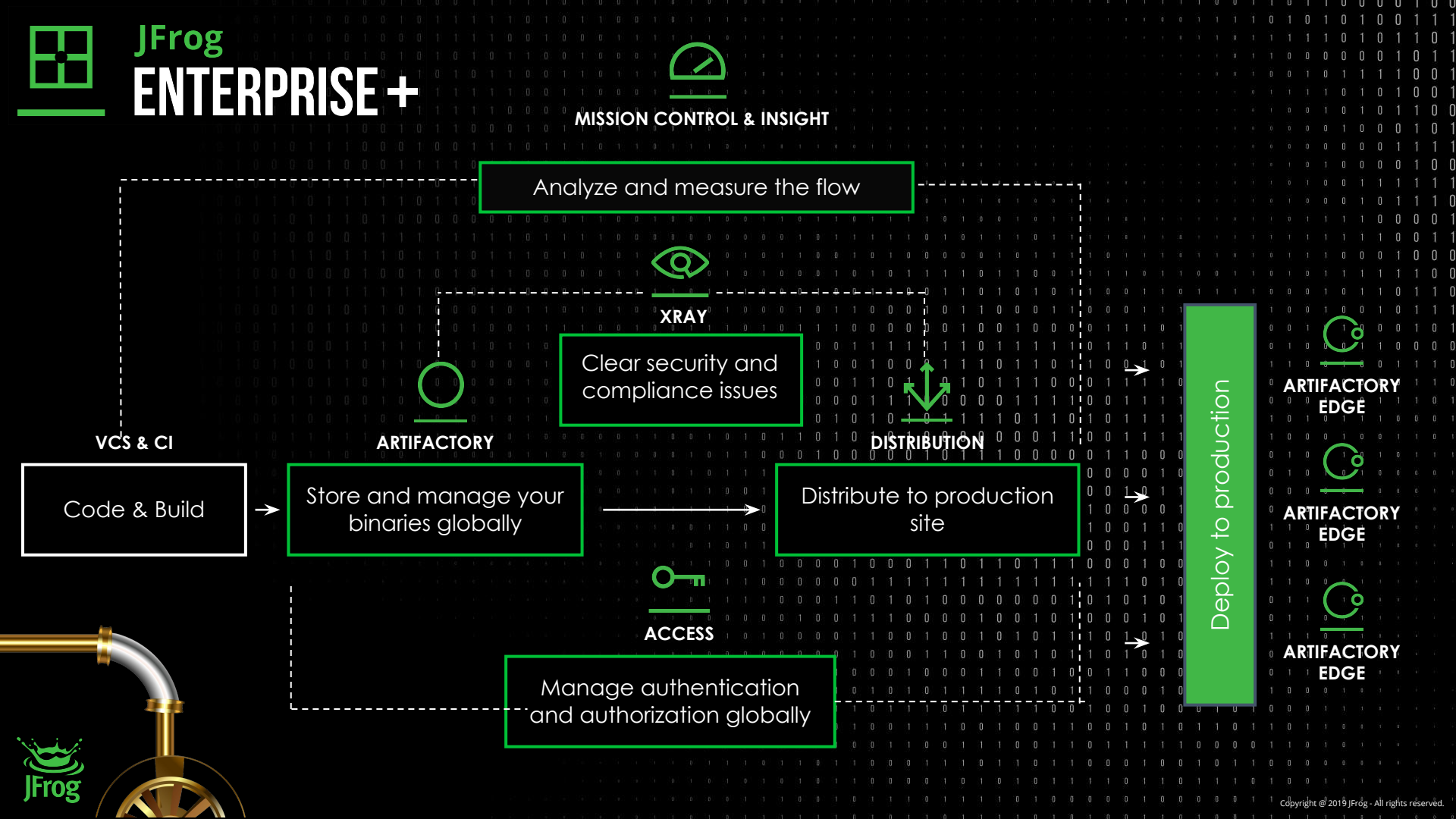
INTEGRATED ECOSYSTEM



HONORED TO LEAD

Internet & Software	Google, vmware, amazon, IBM, salesforce, ORACLE, YAHOO!, slack
Technology & Electronics	LinkedIn, intel, cisco, EMC², CHEF, box, ERICSSON
Banking & Finance	BARCLAYS, Fidelity INVESTMENTS, CHASE, citi, Bank of America, CREDIT SUISSE, WELLS FARGO
Engineering & Aerospace	ANSYS, GM, NASA, AIRBUS DEFENCE & SPACE, BOSCH, HALLIBURTON, GE, BOEING
Retail & Consumer	COSTCO WHOLESALE, NETFLIX, GAP OUTLET, VOLVO, macy's, Riot GAMES
Education & Research	Yale, OHIO UNIVERSITY, Stanford University, University of San Diego, OICR, MIT

JFrog



AGENDA

- Introductions
- Setup Environment
- Artifactory Internals
- *Artifactory Logging*
- *High HTTP Request Throutput*
- *Database Connections Exceeded*
- *Help, I've Got 1 GB Disk Left*
- If time permits

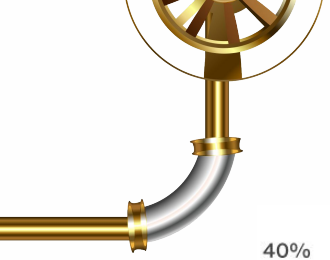
  - JVM Memory Issues

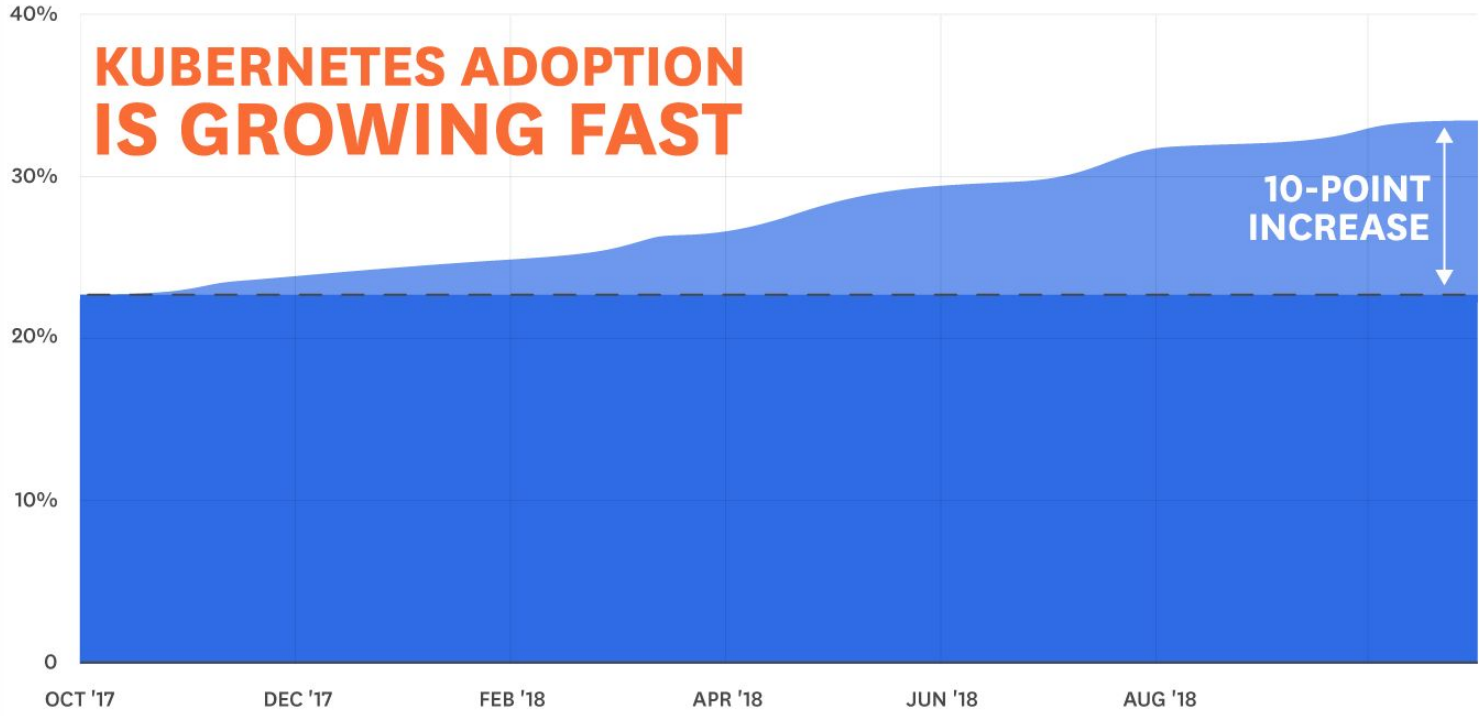
  - Monitoring REST API

Short introduction - Name, Company, Role

SOME BACKGROUND

lecture time

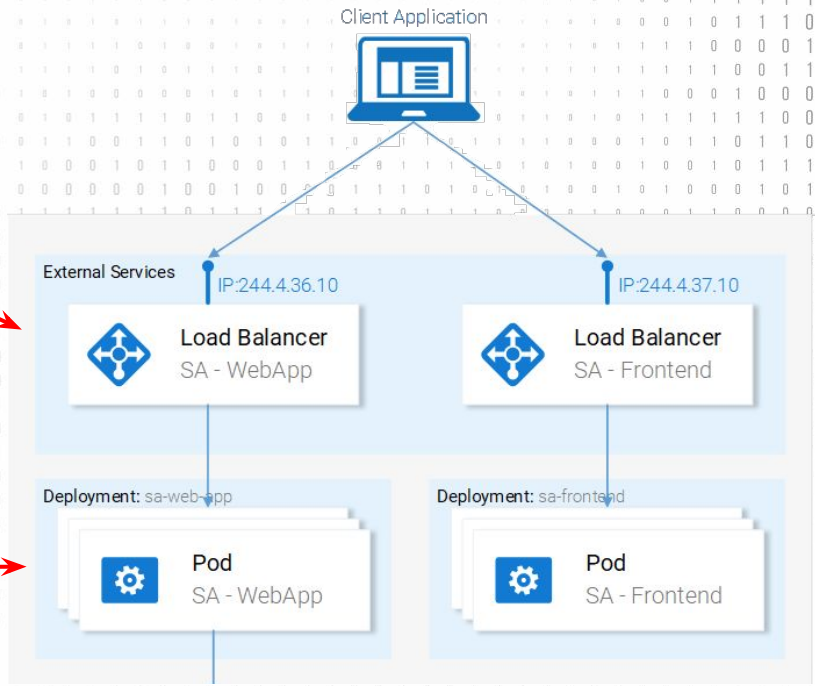




K8S BASIC INTRODUCTION

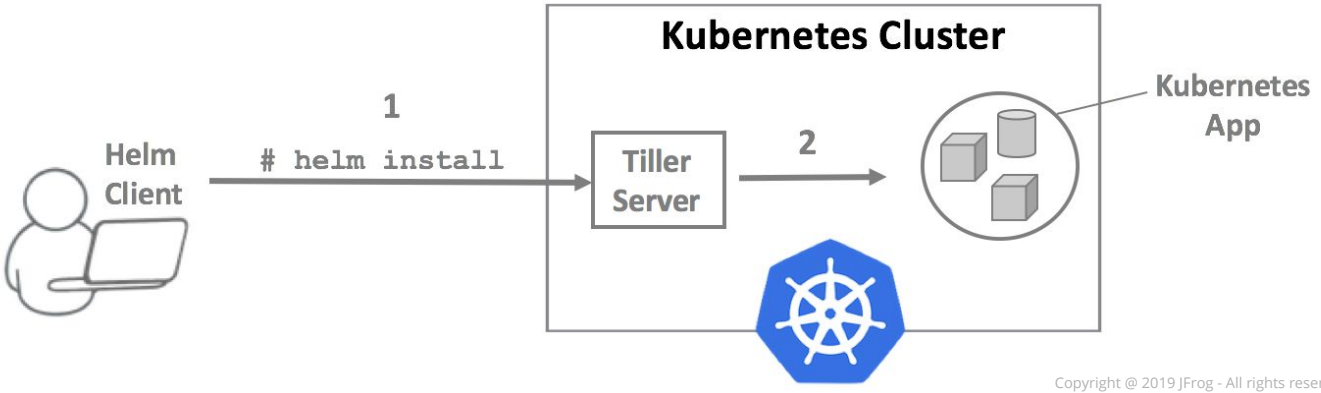
- **Service** is an abstraction which defines a logical set of **Pods** and a policy by which to access them.
  Also called a micro-service.

- **Pod** is a group of one or more **containers** with shared storage/network, and a specification for how to run the containers.

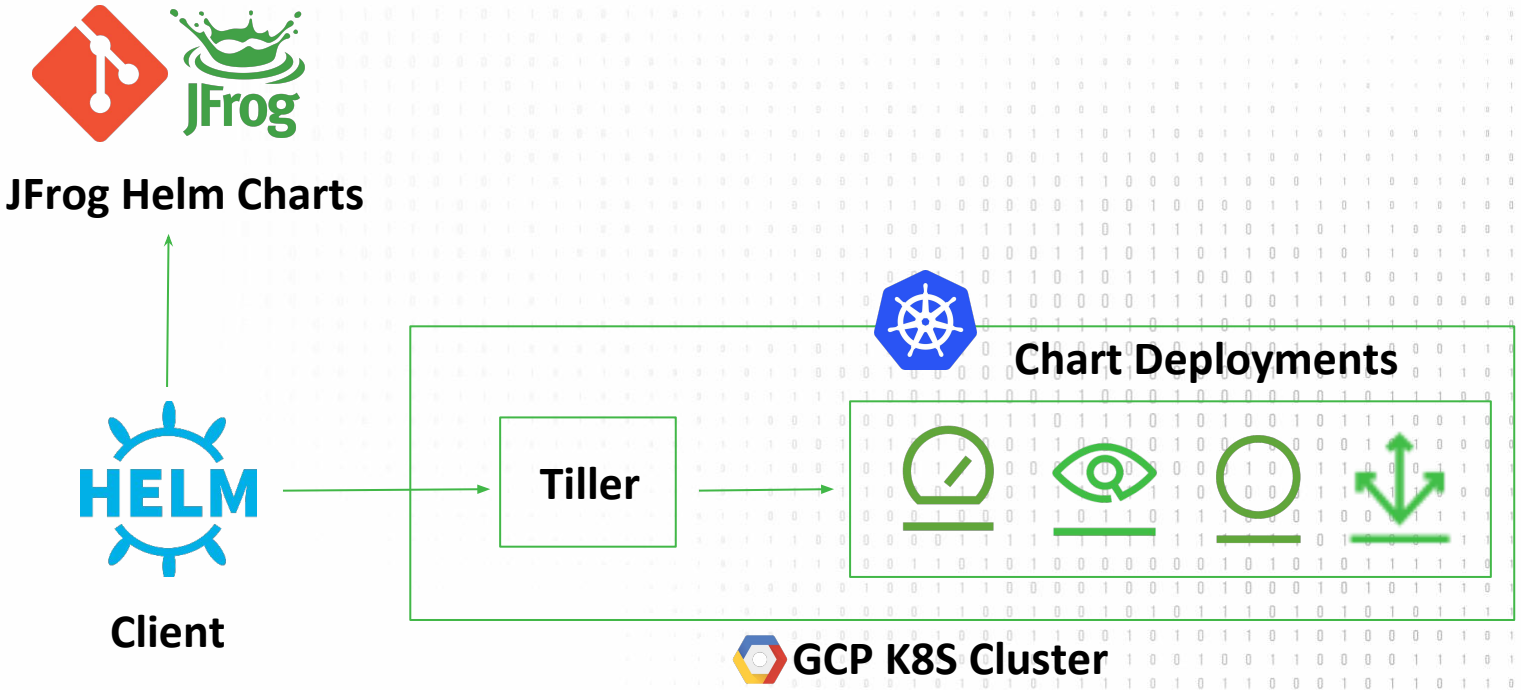


WHAT'S HELM BRING US ?

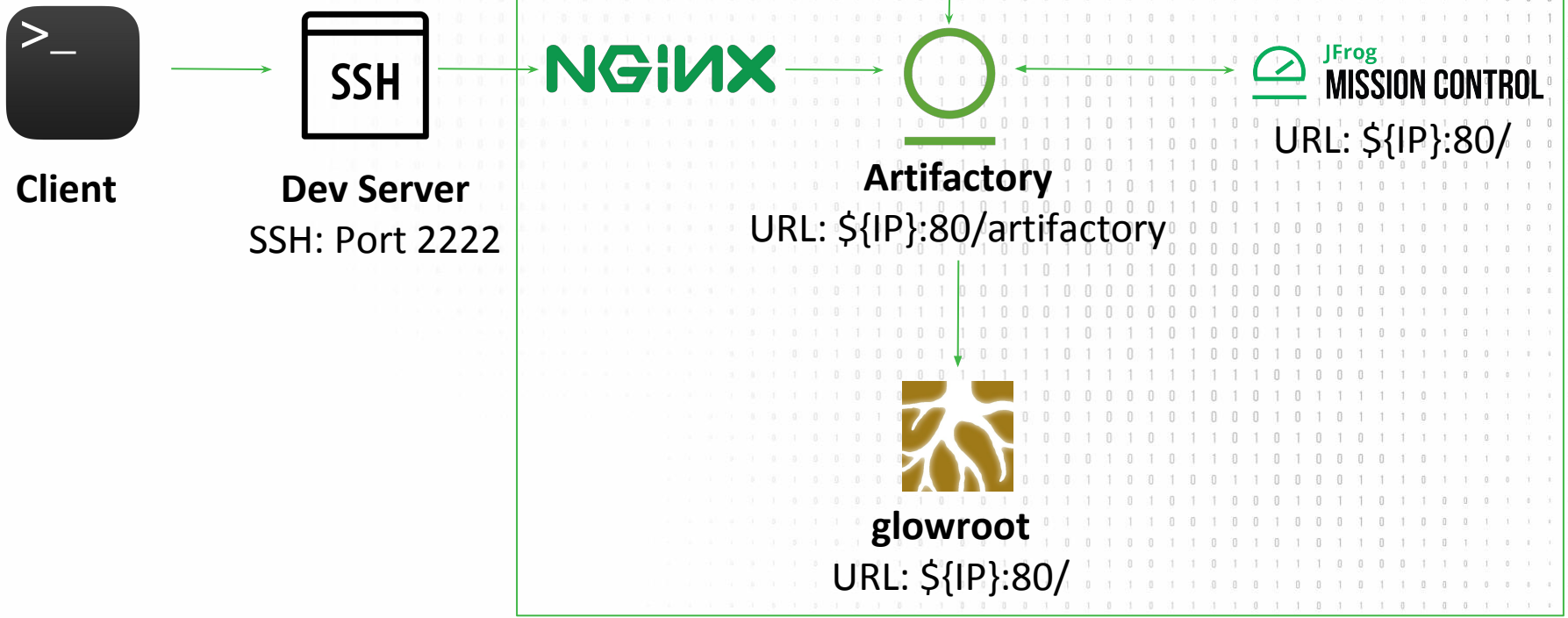
- The Package Manager for Kubernetes
- CLI (helm) and Server component (tiller)
- Wrapper script (Context, Application lifecycle management)
- Dependency management
- Conventions (filename, folder name)
- Add logic to templates



UNDERSTANDING THE ENVIRONMENT DETAILS



OUR DEMO SETUP



SETUP INSTRUCTIONS

- Get Demo Env IP - http://bit.ly/2XffIgf
  type your full name and get IP + cluster name
  `ssh -p 2222 root@x.x.x.x` ,password - **root**

CLASS ENVIRONMENT - SERVICES

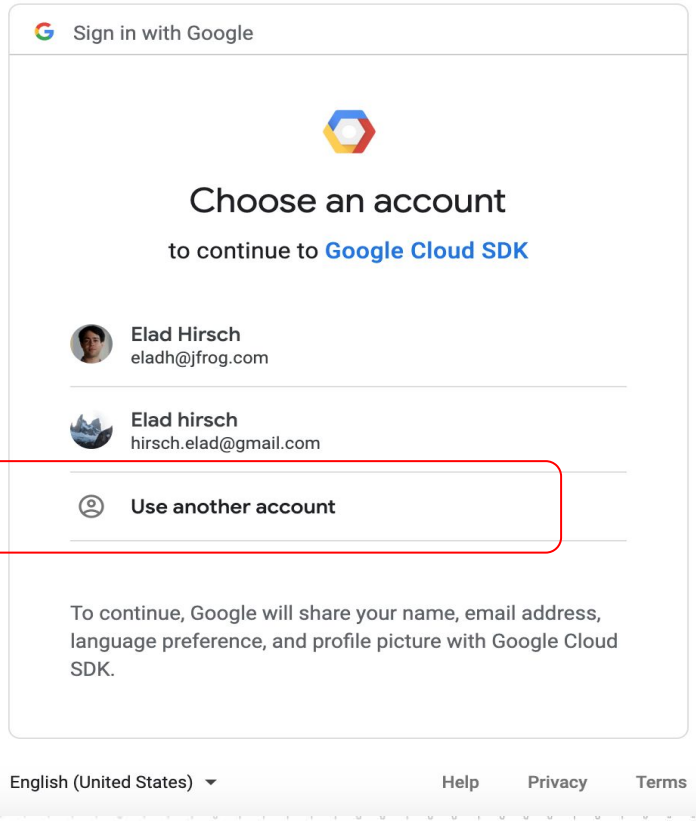
Run `kubectl get svc`

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
artifactory-apm-cassandra	ClusterIP	10.7.240.223	<none>	9042/TCP,9160/TCP	8m
artifactory-apm-cassandra-headless	ClusterIP	None	<none>	7000/TCP,7001/TCP,7199/TCP,9042/TCP,9160/TCP	8m
artifactory-apm-glowroot	LoadBalancer	10.7.246.62	35.238.159.220	80:30414/TCP,8181:30928/TCP	8m
artifactory-artifactory	ClusterIP	10.7.241.198	<none>	8081/TCP	16m
artifactory-artifactory-nginx	LoadBalancer	10.7.240.3	35.225.251.225	80:30160/TCP,443:31396/TCP	16m
artifactory-postgresql	ClusterIP	10.7.253.239	<none>	5432/TCP	16m
jenkins-my-bloody-jenkins	LoadBalancer	10.7.243.58	104.197.138.243	8080:31013/TCP,50000:31611/TCP,16022:31813/TCP	10m
kubernetes	ClusterIP	10.7.240.1	<none>	443/TCP	18m
mission-control	LoadBalancer	10.7.248.201	104.154.178.170	80:31458/TCP,9300:30907/TCP	16m
mission-control-postgresql	ClusterIP	10.7.253.226	<none>	5432/TCP	16m
platform-sshd-dev	LoadBalancer	10.7.255.188	35.202.186.82	2222:31267/TCP	16m
sonarqube-server-postgresql	ClusterIP	10.7.243.200	<none>	5432/TCP	16m
sonarqube-server-sonarqube	LoadBalancer	10.7.251.123	35.232.120.185	9000:30637/TCP	16m

`run /resources/init.sh <clusterName>`

- Open the url and login
  - **User** - swampup2019performance@gmail.com
  - **Password** - zooloo123

```
cluster2-sshd-dev-54bdfdcdd5-9znsc:/resources# ./init.sh  perf-workshop
demo-3-cluster

You are running on a Google Compute Engine virtual machine.
It is recommended that you use service accounts for authentication.

You can run:

 $ gcloud config set account `ACCOUNT`

to switch accounts if necessary.

Your credentials may be visible to others with access to this
virtual machine. Are you sure you want to authenticate with
your personal account?

Do you want to continue (Y/n)?  y

Go to the following link in your browser:

    https://accounts.google.com/o/oauth2/auth?redirect_uri=urn%3Aietf%3
Awg%3Aoauth%3A2.0%3Aoob&prompt=select_account&response_type=code&client
_id=32555940559.apps.googleusercontent.com&scope=https%3A%2F%2Fwww.goog
leapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fa
uth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengin
e.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2
Fwww.googleapis.com%2Fauth%2Faccounts.reauth&access_type=offline

Enter verification code:
```



SETUP INSTRUCTIONS

- Run `kubectl get svc:`

- Locate `artifactory-artifactory-nginx` service `EXTERNAL-IP` (**User** : admin ,**Password** : password)

- Locate `jenkins-my-bloody-jenkins` service `EXTERNAL-IP`, (**User** : admin ,**Password** : password) , **port: 8080**

- Locate `artifactory-apm-glowroot` service `EXTERNAL-IP` (no user/password needed)

CLASS ENVIRONMENT - SERVICES

Run `kubectl get svc`

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
artifactory-apm-cassandra	ClusterIP	10.7.240.223	<none>	9042/TCP,9160/TCP	8m
artifactory-apm-cassandra-headless	ClusterIP	None	<none>	7000/TCP,7001/TCP,7199/TCP,9042/TCP,9160/TCP	8m
artifactory-apm-glowroot	LoadBalancer	10.7.246.62	35.238.159.220	80:30414/TCP,8181:30928/TCP	8m
artifactory-artifactory	ClusterIP	10.7.241.198	<none>	8081/TCP	16m
artifactory-artifactory-nginx	LoadBalancer	10.7.240.3	35.225.251.225	80:30160/TCP,443:31396/TCP	16m
artifactory-postgresql	ClusterIP	10.7.253.239	<none>	5432/TCP	16m
jenkins-my-bloody-jenkins	LoadBalancer	10.7.243.58	104.197.138.243	8080:31013/TCP,50000:31611/TCP,16022:31813/TCP	10m
kubernetes	ClusterIP	10.7.240.1	<none>	443/TCP	18m
mission-control	LoadBalancer	10.7.248.201	104.154.178.170	80:31458/TCP,9300:30907/TCP	16m
mission-control-postgresql	ClusterIP	10.7.253.226	<none>	5432/TCP	16m
platform-sshd-dev	LoadBalancer	10.7.255.188	35.202.186.82	2222:31267/TCP	16m
sonarqube-server-postgresql	ClusterIP	10.7.243.200	<none>	5432/TCP	16m
sonarqube-server-sonarqube	LoadBalancer	10.7.251.123	35.232.120.185	9000:30637/TCP	16m

CLASS ENVIRONMENT - PODS

Run `kubectl get pods`

```
NAME                                          READY   STATUS    RESTARTS   AGE
artifactory-apm-cassandra-0                   1/1     Running   0          12m
artifactory-apm-glowroot-78cccd94c4-7bq7d     1/1     Running   3          12m
artifactory-artifactory-0                     1/1     Running   0          8m
artifactory-artifactory-nginx-7cc965b947-4qj2l   1/1     Running   0          9m
artifactory-postgresql-7497686b49-2hc7h       1/1     Running   0          21m
jenkins-my-bloody-jenkins-67f95fb765-2pjx9    1/1     Running   0          15m
mission-control-0                             5/5     Running   0          21m
mission-control-postgresql-6d6b47f775-b5fgw   1/1     Running   0          21m
platform-sshd-dev-7bd8675fb6-hmqq9            1/1     Running   0          21m
sonarqube-server-postgresql-5d76ddcb75-2k558  1/1     Running   0          21m
sonarqube-server-sonarqube-5ccc686887-6jc7q   1/1     Running   2          21m
```

COMMON COMMANDS

- View all K8S services/pods

```
kubectl get svc / kubectl get pods
```

- Get logs for artifactory pod

```
kubectl logs -f pod_name --container container_name
```

- SSH to Artifactory pod

```
kubectl exec -ti pod_name -- /bin/bash
```

- Change Artifactory pod

```
helm upgrade chart_name jfrog/artifactory --set
artifactory.xxxxxxx=value
```

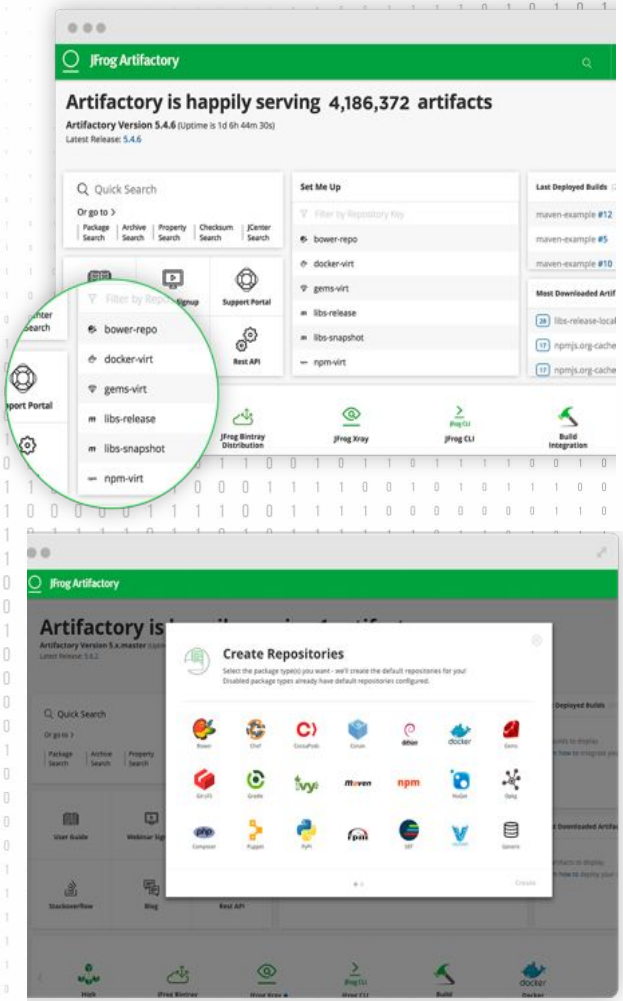
- Install new Articatory instance

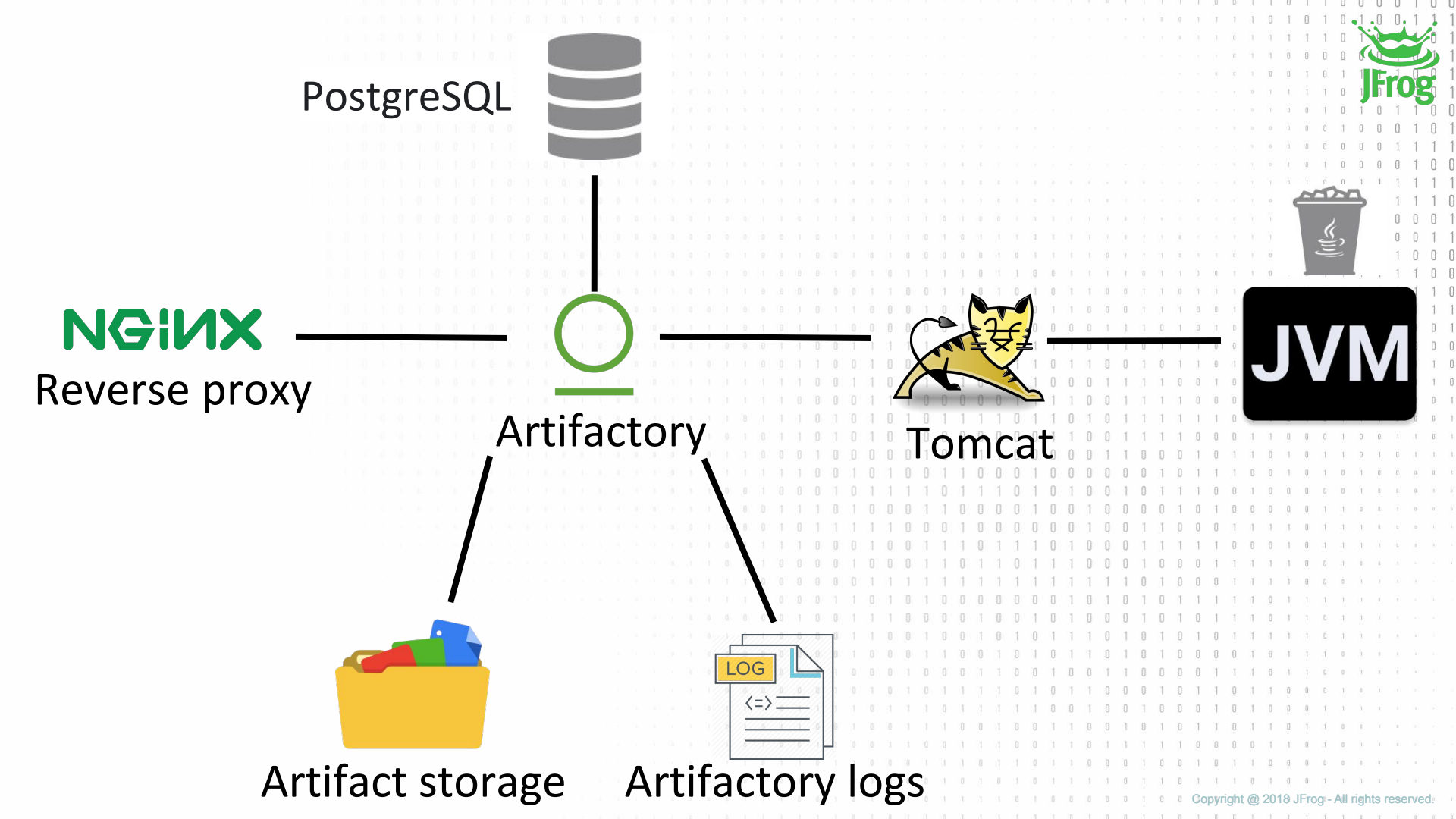
```
helm install --name chart_name jfrog/artifactory
```

ARTIFACTORY INTERNALS

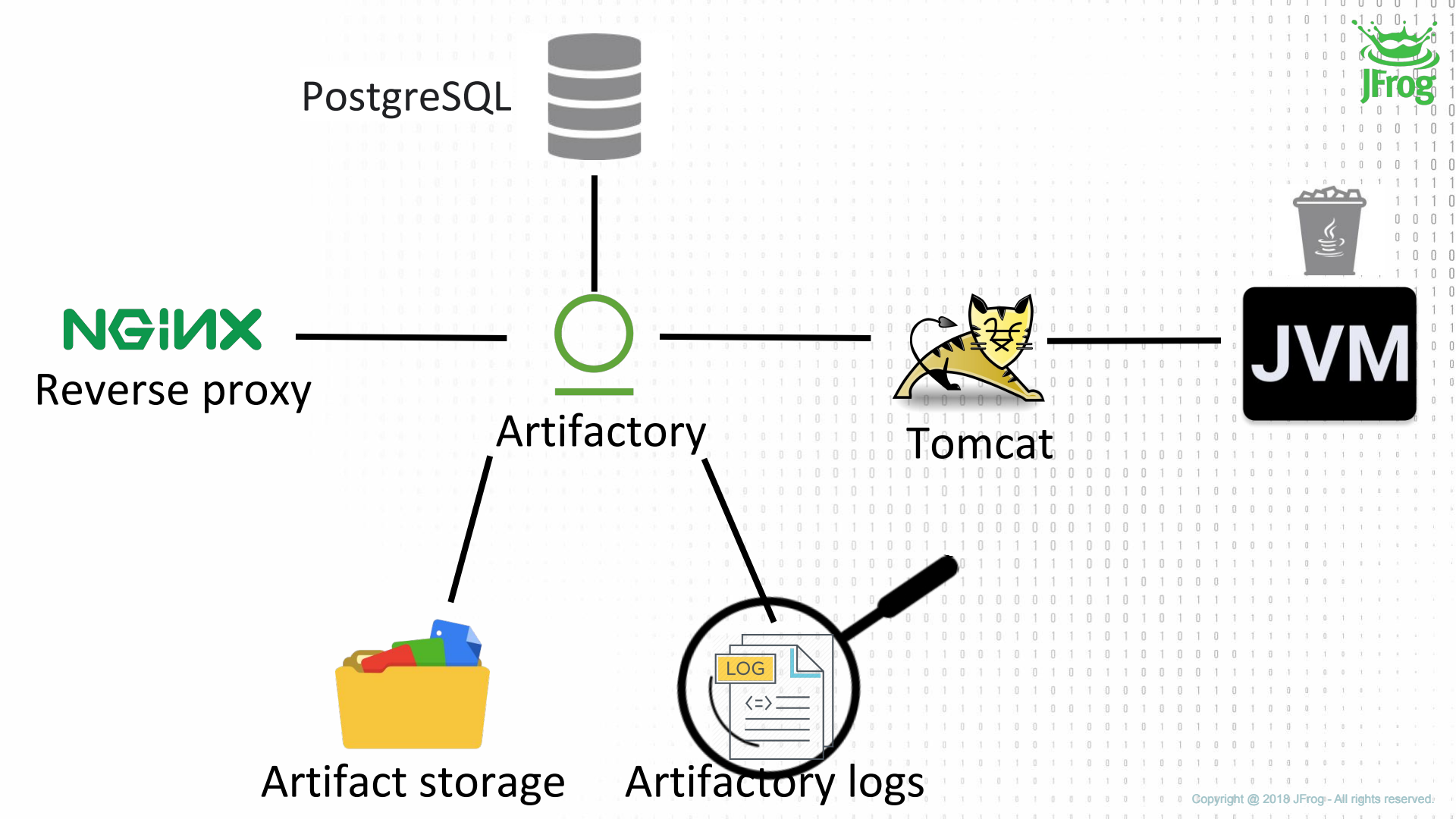
ARTIFACTORY

- Stores and Manages your binaries
- Universal
- Scalable
- Fully Automated
- Integrated with all major build tools and CI servers





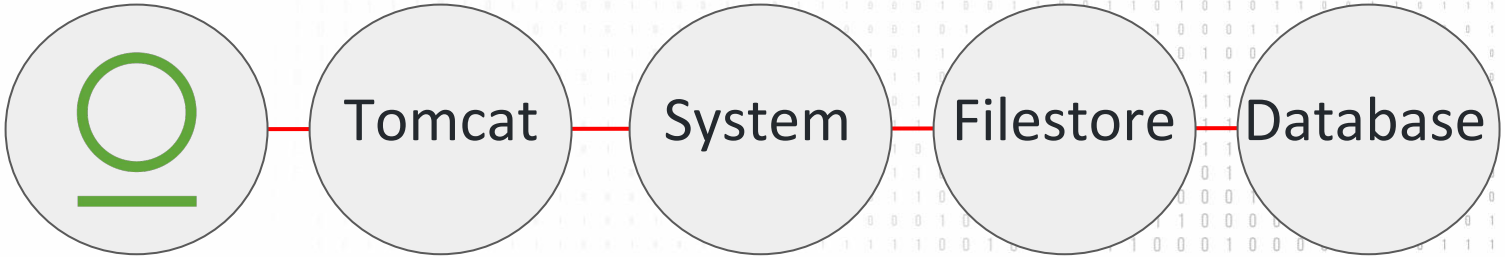
Labs details - http://bit.ly/2XQZ1Vs



LOVE YOUR LOGS, IT PAYS OFF!

EVENT SOURCES

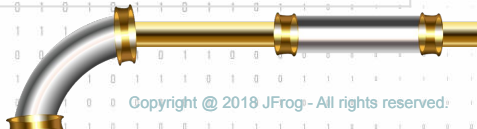
- Logs primary source



- Health & Readiness check available via REST API: `/api/system/ping`

ARTIFACTORY LOGS: `logback.xml`

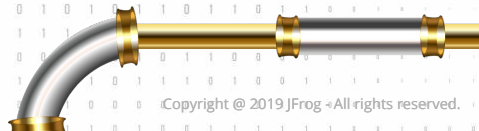
`artifactory.log`	The main Artifactory log file containing data on Artifactory server activity.
`access.log`	Access log containing important information about accepted and denied requests, configuration changes and password reset requests. The originating IP address for each event is also recorded.
`request.log`	Generic http traffic information similar to the Apache HTTPd request log.
`request_trace.log`	Extended information, including HTTP headers, when requests are made including `?trace`
`import.export.log`	Log used for tracking the process of long-running import and export commands.
`traffic.log`	Traffic log providing information similar to the request log, specifically for replication. Disabled by default. To enable, in the set `artifactory.traffic.collectionActive=true` in `artifactory.system.properties`



LAB1: UPDATING LOGGING LEVELS

ACCESS

- Bundled with Artifactory and runs as a separate service under the same Tomcat
- Provides a common authentication and authorization infrastructure for all of JFrog products to manage users, groups, permissions and tokens
- Different access services can be federated into a single “circle of trust”
  - Enables synchronization of users, groups and permissions between sites rather than having to define them separately
  - Enables single-sign-on for all JFrog services



LAB1: UPDATING LOGGING LEVELS

- Send an authenticated request to Artifactory with **bad credentials**, i.e:
  `curl -u admin:wrongPassword http://x.x.x.x/artifactory/api/system/ping`
- Notice the 401, but nothing in the **/var/opt/jfrog/artifactory/logs/artifactory.log**
- Let's use the Artifactory HTTPClient debug logging
- Find the relevant communication attempt between Artifactory<->Access, and Access return **401 response** to Artifactory

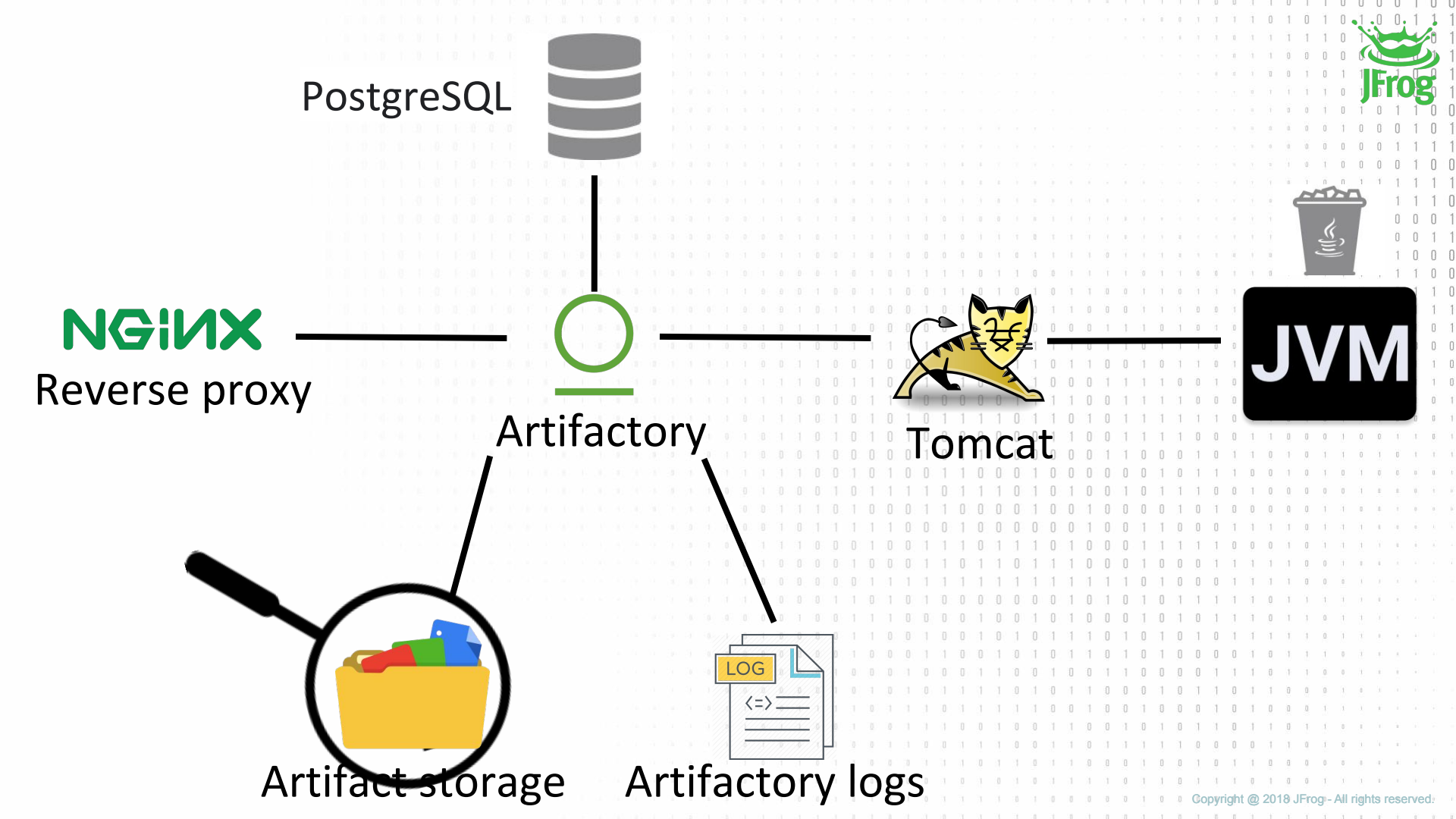
Apache HTTPClient - *org.apache.http.wire*

- The wire log is used to log all data transmitted to and from servers when executing HTTP requests.
- Enable HTTPClient debug logging:

```
<logger name="org.apache.http.wire”>
        <level value="debug"/>
        </logger>
```

This log should only be enabled to debug problems, as it will produce an extremely large amount of data

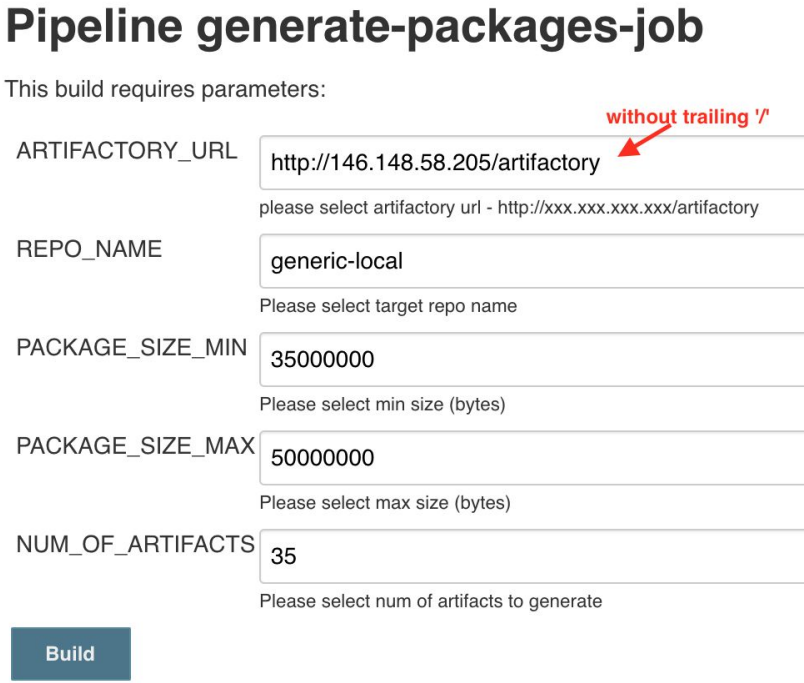
LAB2 SETUP: HELP, I'VE GOT 1GB DISK LEFT!



LAB2: HELP, I'VE GOT 1GB DISK LEFT!

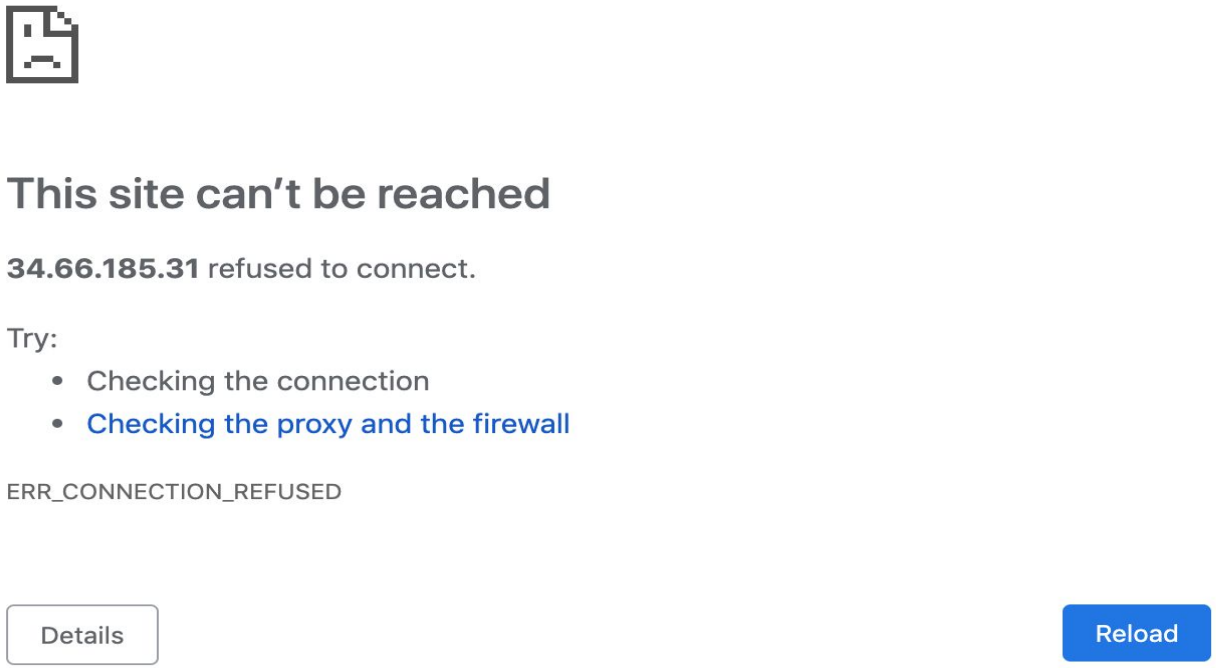
OK, not really 1GB disk left, but, let's create a lot of extra artifacts

- Go to Jenkins http://xxx.xxx.xxx.xxx:8080
- Click on the **generate-packages-job**
- Run once to trigger Job Param
- The build will fail
- Click once on "Build with Parameters"
- Create a generic repository on Artifactory
- **REPO_NAME** - The name of the repo
- Artifactory URL need to be : http://xxx.xxx.xxx.xxx/**artifactory**
- Enter the parameters and click "Build"
- **Leave this job running**



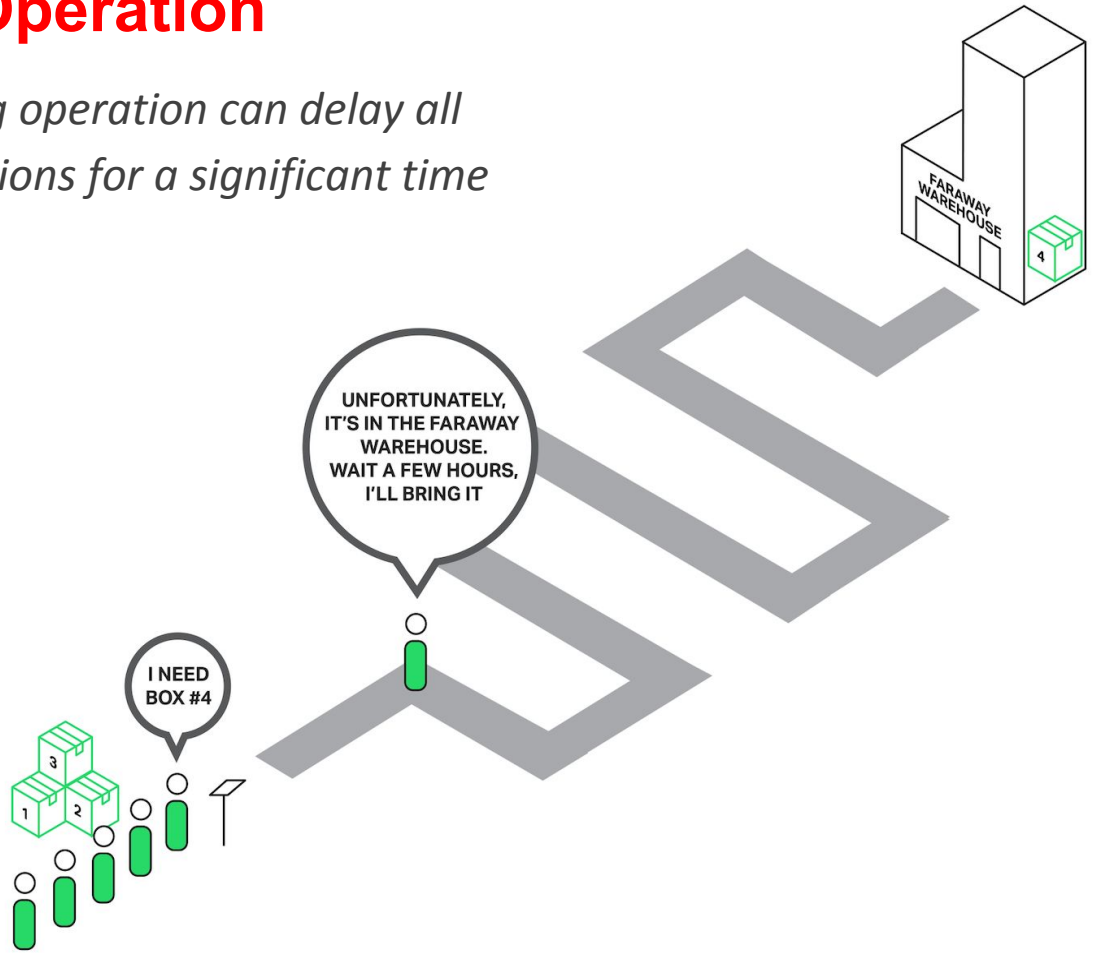
LAB3: HIGH HTTP(S) REQUESTS

LAB3: HIGH HTTP(S) REQUESTS



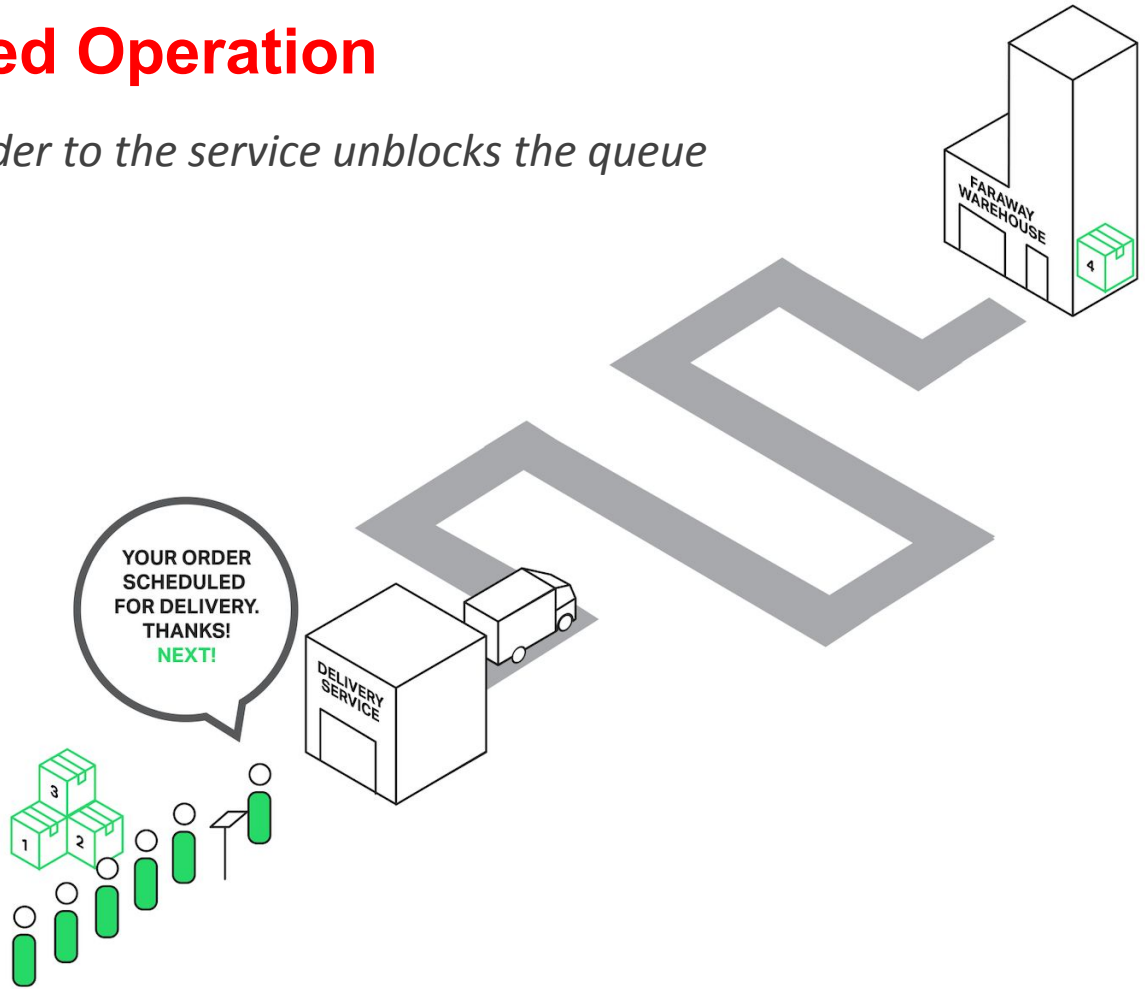
Blocking Operation

Just one blocking operation can delay all following operations for a significant time

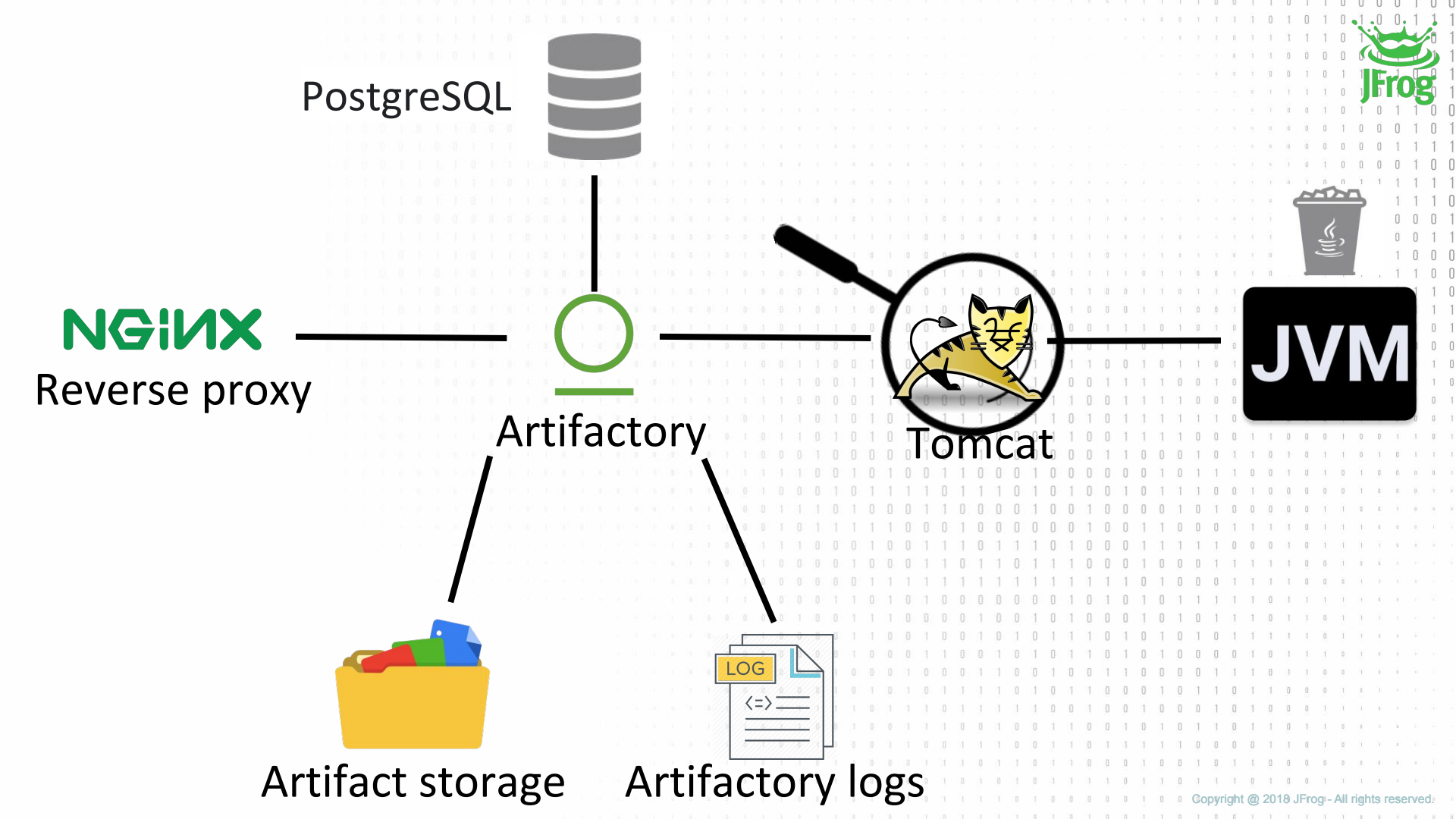


Scheduled Operation

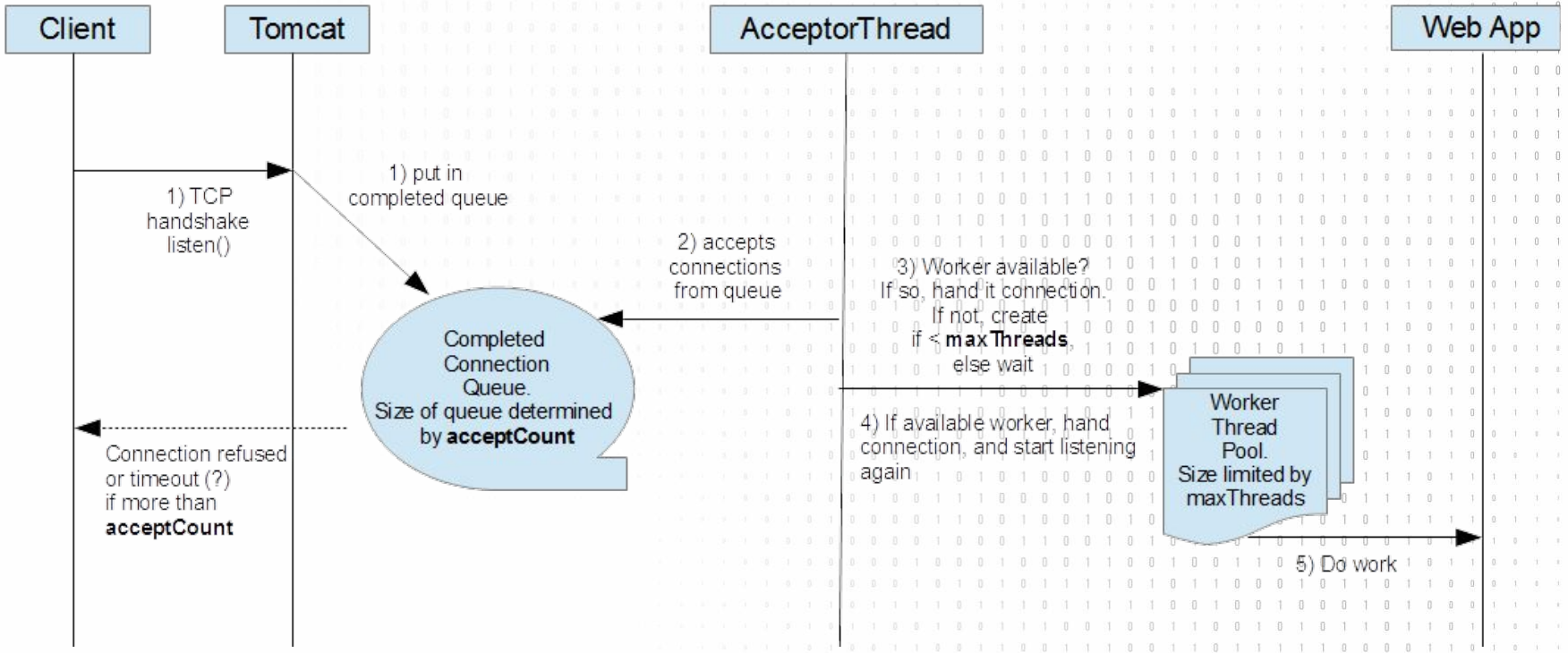
Passing an order to the service unblocks the queue



BUT STILL - HOW LONG IS THE LINE?



<Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000"
redirectPort="8443" **acceptCount="1000" maxConnections="500"** />

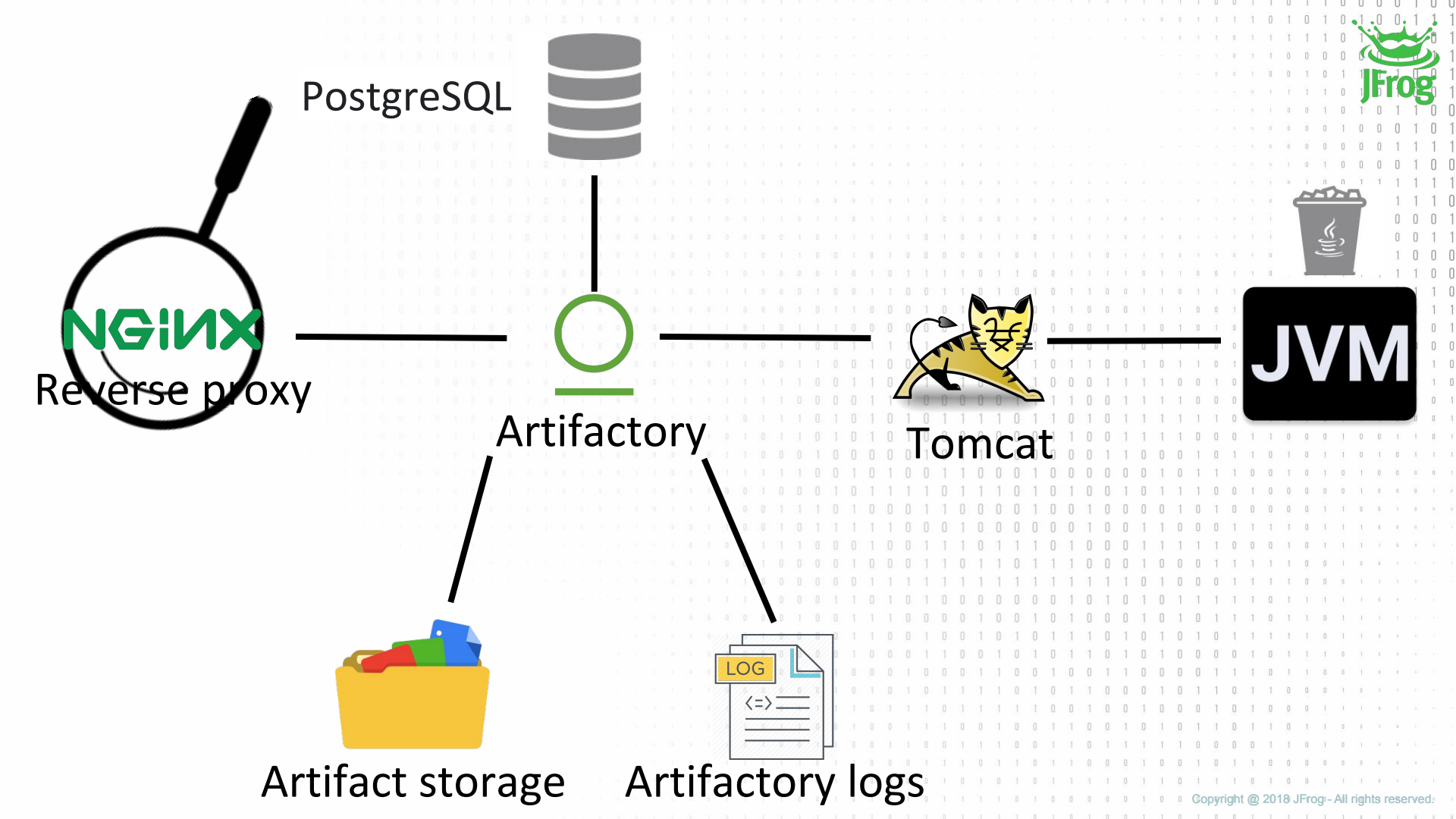


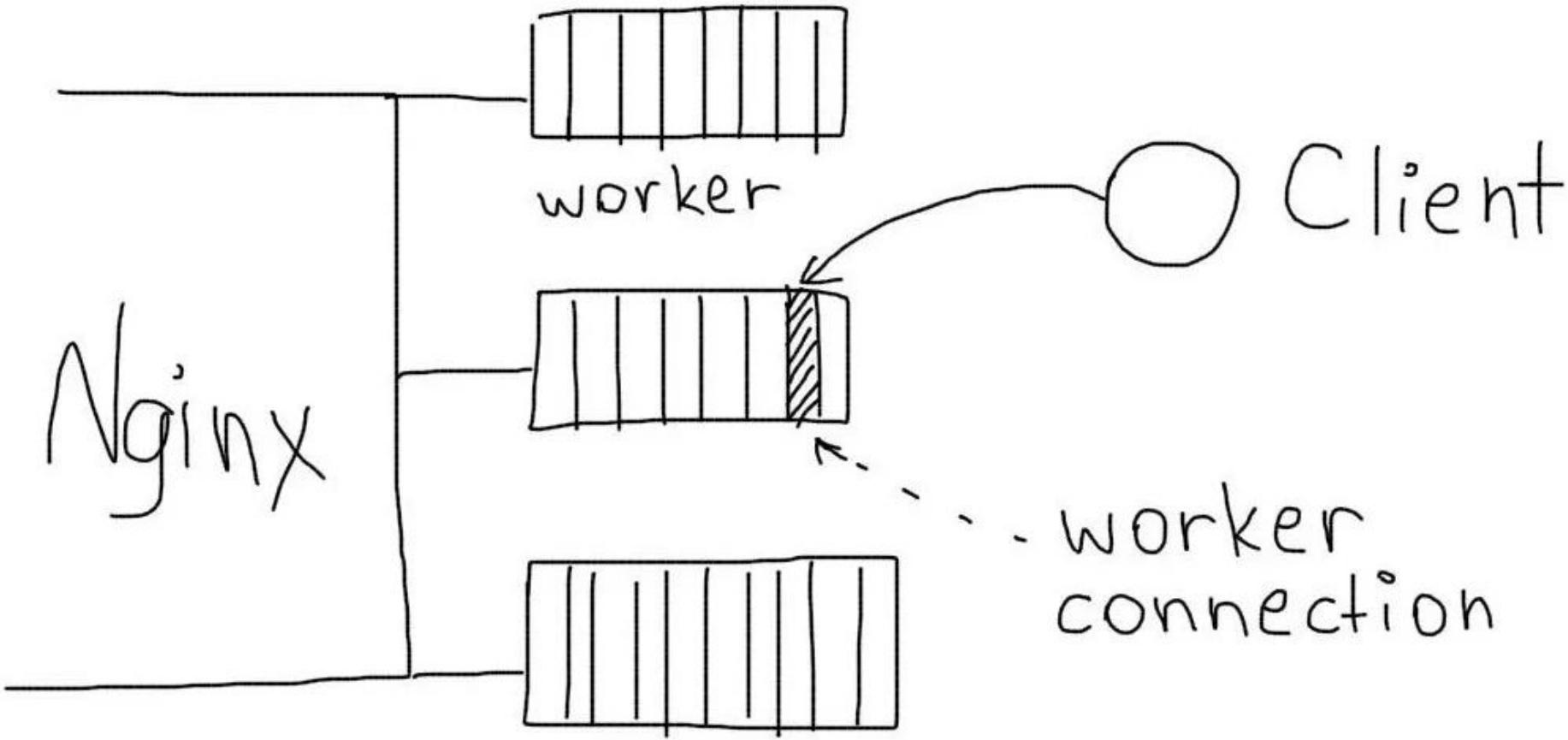
SET ARTIFACTORY MAX THREADS

- https://github.com/jfrog/charts/tree/master/stable/artifactory
- https://www.jfrog.com/confluence/display/RTF/Installing+with+Docker
  (look for **SERVER_XML_ARTIFACTORY_MAX_THREADS**)

```
helm upgrade artifactory jfrog/artifactory
--version 7.14.3 -f artifactory.yaml
```

```
artifactory:
  extraEnvironmentVariables:
   - name: SERVER_XML_ARTIFACTORY_EXTRA_CONFIG
     value: 'maxConnections="2" acceptCount="1" '
   - name: SERVER_XML_ARTIFACTORY_MAX_THREADS
     value: "2"
```

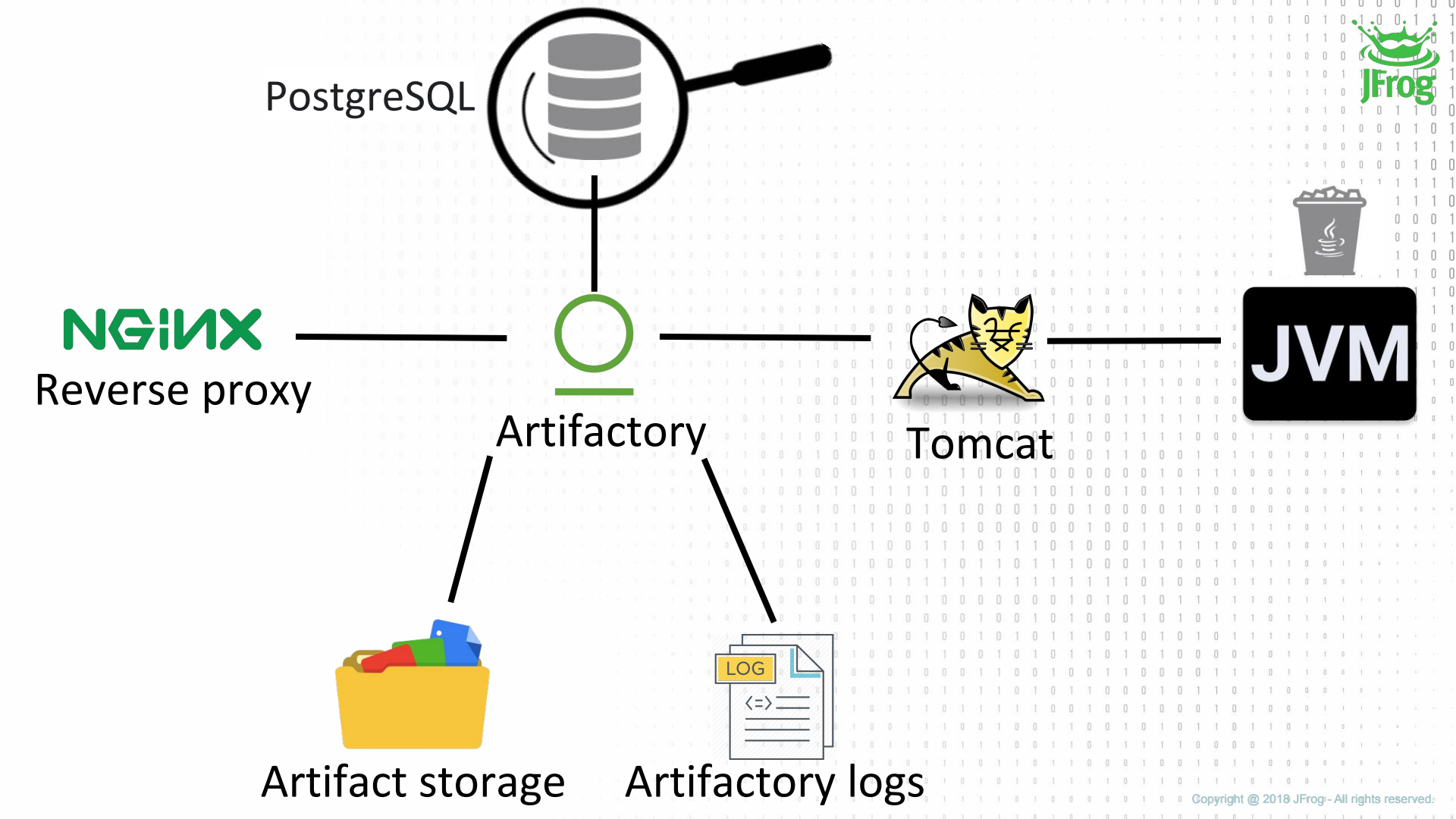




SET NGINX WORKER CONNECTION

- Create a configMap: `kubectl create configmap nginx-conf --from-file=nginx.conf`
- reload artifactory: `helm upgrade artifactory jfrog/artifactory --version 7.14.3 --set nginx.customConfigMap=nginx-conf`
- How it's affecting our server - try running ab command to simulate traffic
- Change **worker_connections** by increasing this value, the hope is that we can increase the capacity of each worker process

LAB4: DATABASE CONNECTIONS EXCEEDED



LAB4: DATABASE CONNECTIONS EXCEEDED

- Change Postgres settings:

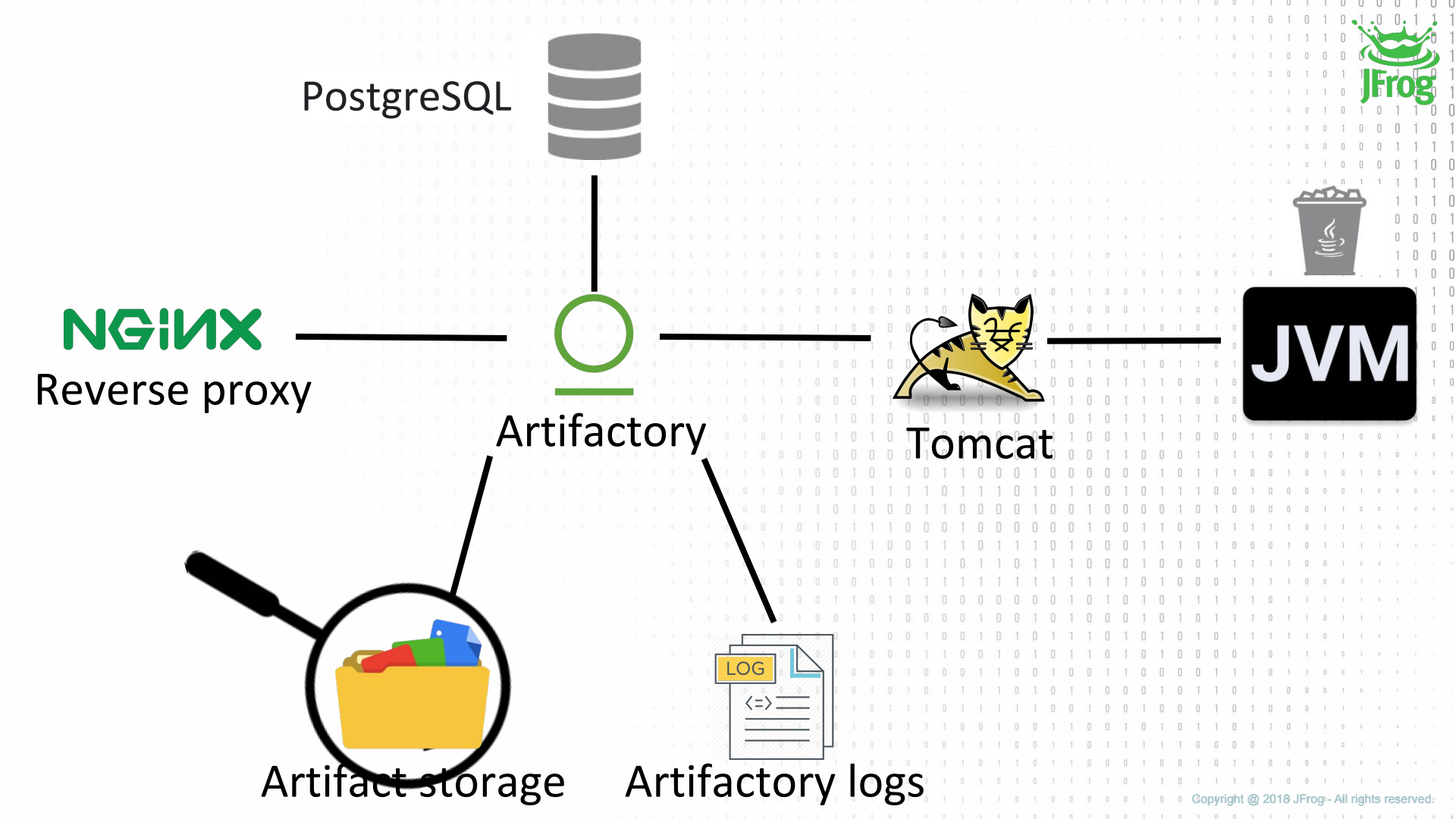
```
helm upgrade artifactory jfrog/artifactory
 --version 7.14.3
 --set postgresql.postgresPassword=zooloo
--set postgresql.postgresConfig.max_connections=1
--set
postgresql.postgresConfig.superuser_reserved_connections=0
```

- Go to our Artifactory instance
- Notice that it hangs and doesn’t load completely

LAB4: DATABASE CONNECTIONS EXCEEDED

- In the **artifactory.log** file you should see something like this:
  - `Caused by: org.postgresql.util.PSQLException: Data source rejected establishment of connection, message from server: "Too many connections"`
- Can we resolve this issue by adjusting Artifactory configuration only?
  https://jfrog.com/knowledge-base/how-do-i-tune-artifactory-for-heavy-loads/

LAB2 PHASE II: HELP, I'VE GOT 1GB DISK LEFT!



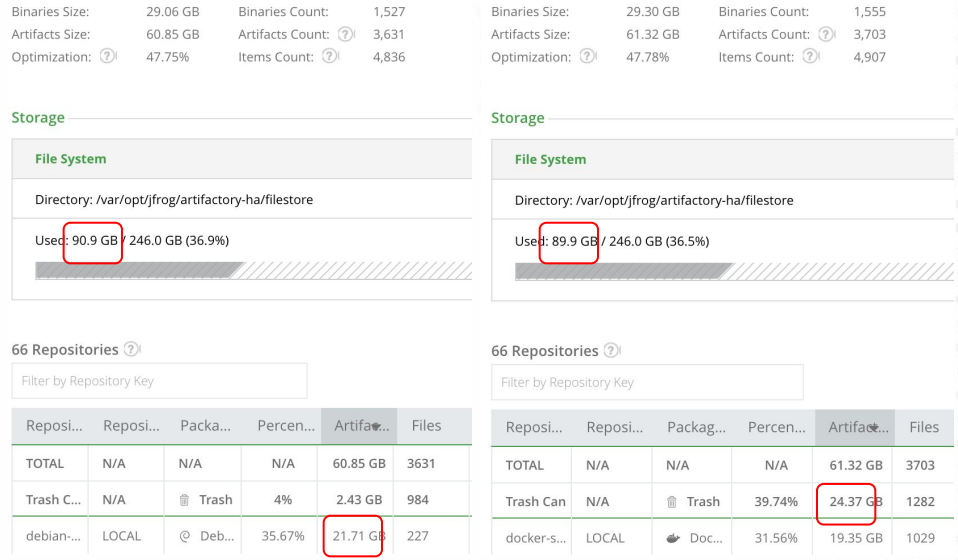
LAB2: HELP, I'VE GOT 1GB DISK LEFT!

- If you have implemented a promotion policy you might be able to delete your snapshot repo or equivalent in entirety
- Take a more proactive approach to managing it using a combination of plugins, AQL and the CLI
- Schedule deletes

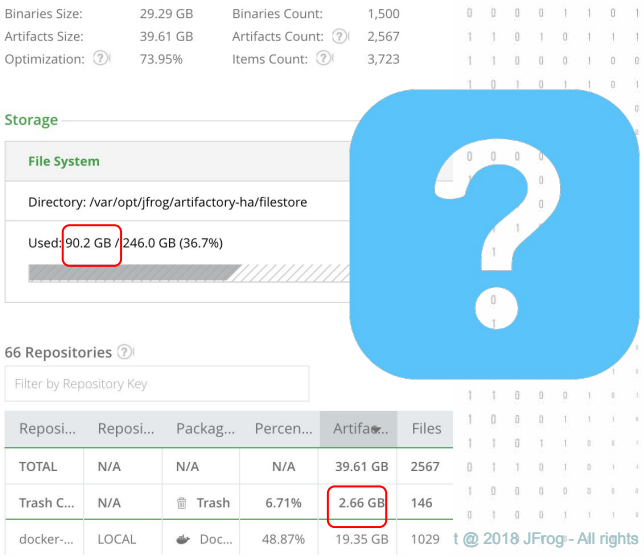
LAB2: HELP, I'VE GOT 1GB DISK LEFT!

- Browse to the **repo we created** earlier, right click on it and click "Delete Content"
- Keep the storage window open before you delete, and open a new one after
- As you can see - the **storage usage is the same** ever after we delete content

before



after



ARTIFACTORY - GARBAGE COLLECTION

- When an Artifactory user "deletes" a file, what is actually deleted is the reference from the Artifactory database to the physical file.
- Before actually deleting a file Artifactory ensure that there are no other references to the file.

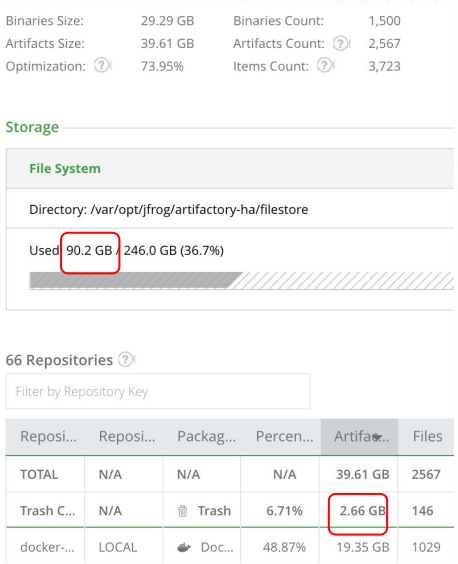
Scanning the system is very CPU intensive

Schedule based “Garbage Collection”

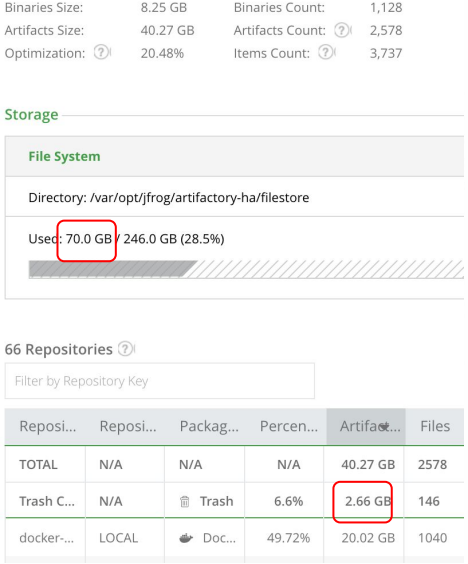
LAB2: HELP, I'VE GOT 1GB DISK LEFT!

- Now that we know what GC does, let's run it by going to:
  WebUI → Admin → Advanced → Maintenance → Garbage Collection → Run

before



after



STORAGE RELATED REST API

- Delete artifact (REST API) :
  - DELETE http://xxx.xxx.xxx.xxx/artifactory/repoName/path
- Empty Trash Can (REST API) :
  - DELETE /api/trash/clean/repoName/path

Advanced cleanup using AQL and JFrog CLI

https://www.jfrog.com/confluence/display/RTF/Artifactory+Query+Language

Let’s build an AQL query that identifies candidate artifacts for cleanup with **size > 45 MB** and invoke it with JFrog CLI

DATA COLLECTION (MONITORING) POINTS

FAULT - EVENT SOURCES

- REST API: `/api/system/ping`
- Provides a complete system health check
  - Returns an HTTP 200 code with “OK” if Artifactory is running
  - Otherwise returns a 5xx with stated reason
  - Checks to see if socket is open, and if full Artifactory service including DB and filestore is functional

MONITORING REST API

Name	Usage	Description	Produces
Scheduled Replication Status	`GET /api/replication/{repoKey}`	Returns the status of scheduled cron-based replication jobs define via the Artifactory UI on repositories. Supported by local, local-cached and remote repositories	`application/json`
Get Storage Summary Info	`GET /api/storageinfo`	Returns storage summary information regarding binaries, file store and repositories.	`application/json`
System Info	`GET /api/system`	Display different system information such as JVM runtime parameters, JVM arguments, memory usage and more.	`text/plain`

LAB5 : MONITOR ARTIFACTORY JVM

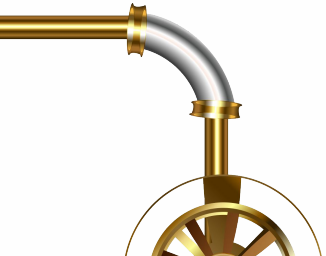
GlowRoot APM

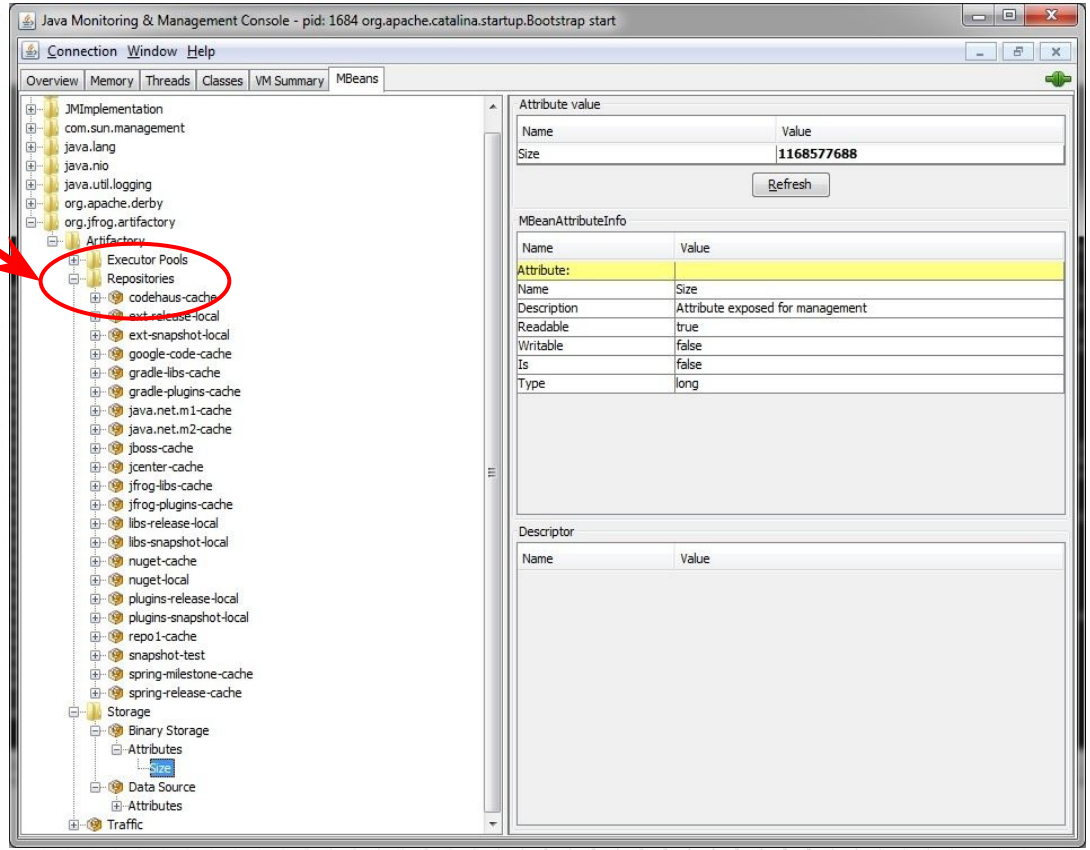
- Community driven open source monitoring for JVM based applications
- Capture errors, interact with JVM, identify bottlenecks and provide off the shelf monitoring
- Single instance or central collector configuration available
- Multi node architecture uses Cassandra for data-store with a **docker** and **Helm** deployments available
- GlowRoot url - look for the ***artifactory-apm-glowroot*** service

ARTIFACTORY MBeans

Artifactory exposes MBeans, under the *org.jfrog.artifactory* domain

Let you monitor repositories, executor pools, storage and HTTP connection pools





Kahoot!

JFrog Certification

[Certification FAQs]

[Study Guide]

JFrog Certification

Ready, Cert, Go!

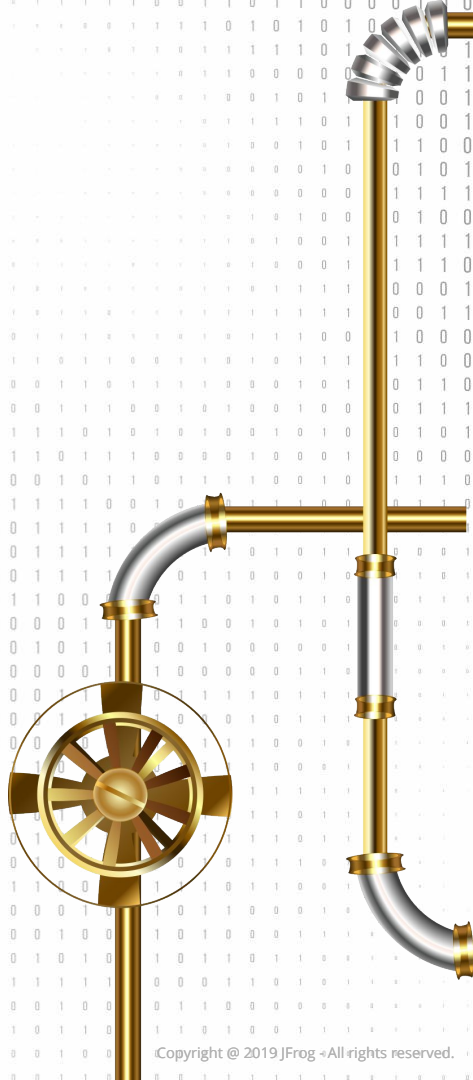
Everyone gets a voucher by email

90 minutes proctored exam - requires a webcam

Practical Knowledge

For more information please contact

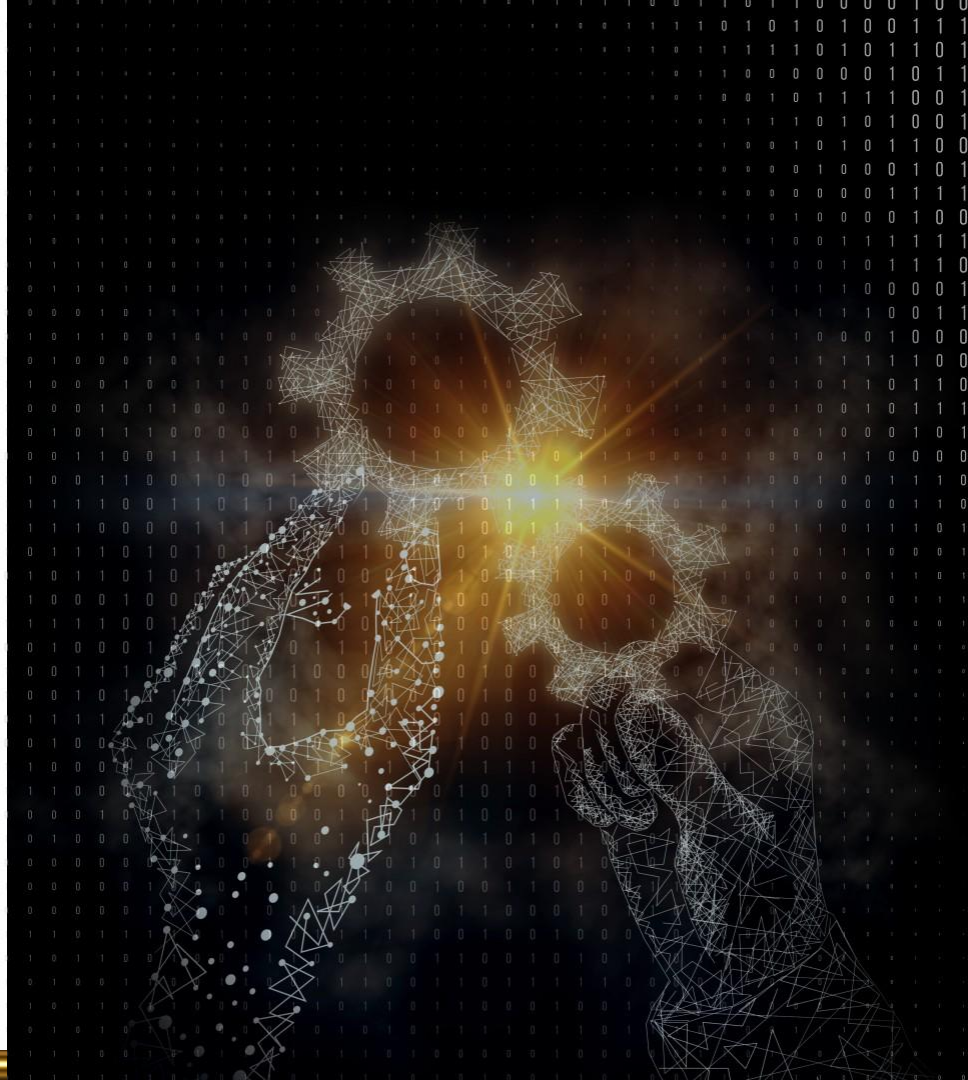
certification@jfrog.com



JFrog Certification

Ready, Cert, Go!

THANK YOU!



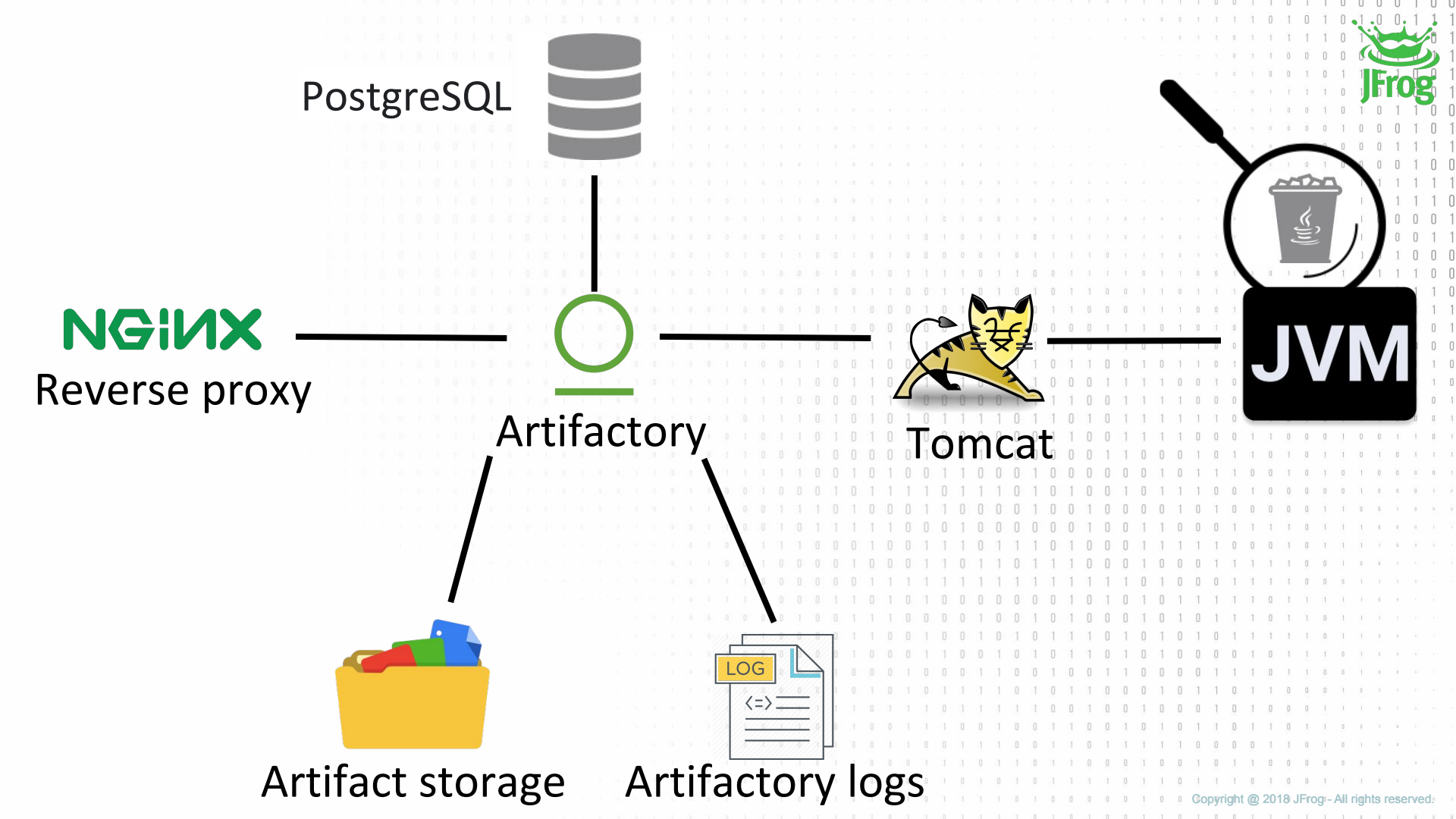
FAULT - LOG FILES

Appender	Level	File Name	Message Pattern
CONSOLE	warn	ARTIFACTORY_HOME/logs/catalina/catalina.out	%date ${artifactory.contextId}[%thread] [%-5p] \(%-20c{3}:%L\) - %m%n
FILE	warn	ARTIFACTORY_HOME/logs/artifactory.%i.log	%date [%thread] [%-5p] \(%-20c{3}:%L\) - %m%n
ACCESS	info	ARTIFACTORY_HOME/logs/access.%i.log	%d %m%n
IMPORT.EXPORT	DEBUG	ARTIFACTORY_HOME/logs/import.export.%i.log	%date [%-5level] \(%logger{32}:%line\) %message%n
TRAFFIC	info	ARTIFACTORY_HOME/logs/traffic.%d{yyyyMMdd}	%message%n
REQUEST	info	ARTIFACTORY_HOME/logs/request.%i.log	%message%n
REQUEST_TRACE	info	ARTIFACTORY_HOME/logs/request_trace.%i.log	%date %message%n
JFROG_ACCESS	info	ARTIFACTORY_HOME/access/logs/jfrog_access.%i.log	%date [%thread] [%-5p] \(%-20c{3}:%L\) - %m%n
JFROG_ACCESS_AUDIT	info	ARTIFACTORY_HOME/access/logs/audit.%i.log	%date %message%n
SHA256_MIGRATION	info	ARTIFACTORY_HOME/logs/sha256_migration.%i.log	%date ${artifactory.contextId}[%thread] [%-5p] \(%-20c{3}:%L\) - %m%n
PATH_CHECKSUM_MIGRATION	info	ARTIFACTORY_HOME/logs/path_checksum_migration.%i.log	%date ${artifactory.contextId}[%thread] [%-5p] \(%-20c{3}:%L\) - %m%n

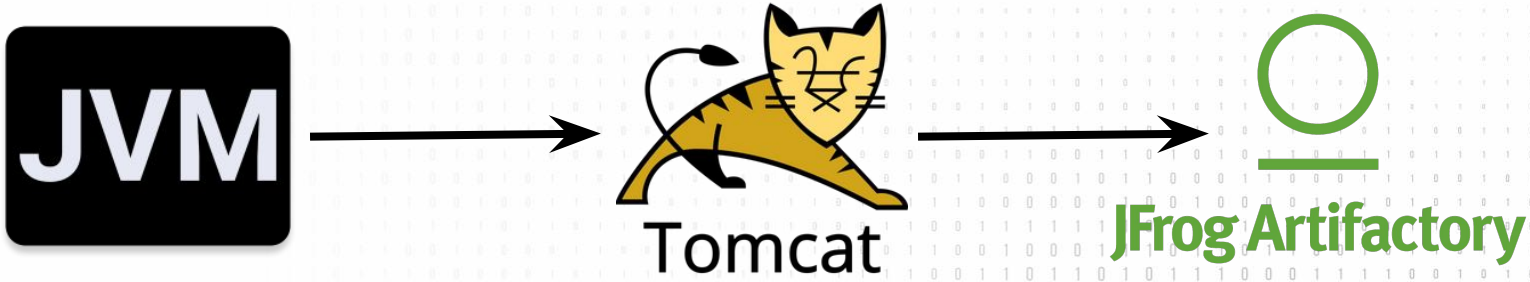
FAULT - TOMCAT LOGS: `ARTIFACTORY_HOME/logs/catalina/`

`catalina.out`	Stdout/Stderr stream for Tomcat's JVM. Default destination for all Tomcat log statements. May include uncaught exceptions printed by `java.lang.ThreadGroup.uncaughtException(..)` or thread dumps, if requested via a system signal. This is the next place to check ***after*** Artifactory logs if something is wrong.
`localhost.<date>.log`	The log of the virtual host within Tomcat that runs the web application container. Normally does not contain much of interest but would be the ***next*** place to check after `catalina.out`.
`catalina.<date>.log`	The web application container log file.
`host-manager.<date>.log` `manager.<date>.log`	The logs of the related web applications. • `host-manager` for Virtual hosts • `manager` for the deploying and undeploying applications

LAB5: JVM MEMORY ISSUES



TOMCAT - JVM TUNING



-server -Xms512m -Xmx2g -Xss256k -XX:+UseG1GC

By increasing your heap (Xmx) you increase bandwidth capability of Artifactory

16gb of Heap is equal to about 1GBit/s Network Speed

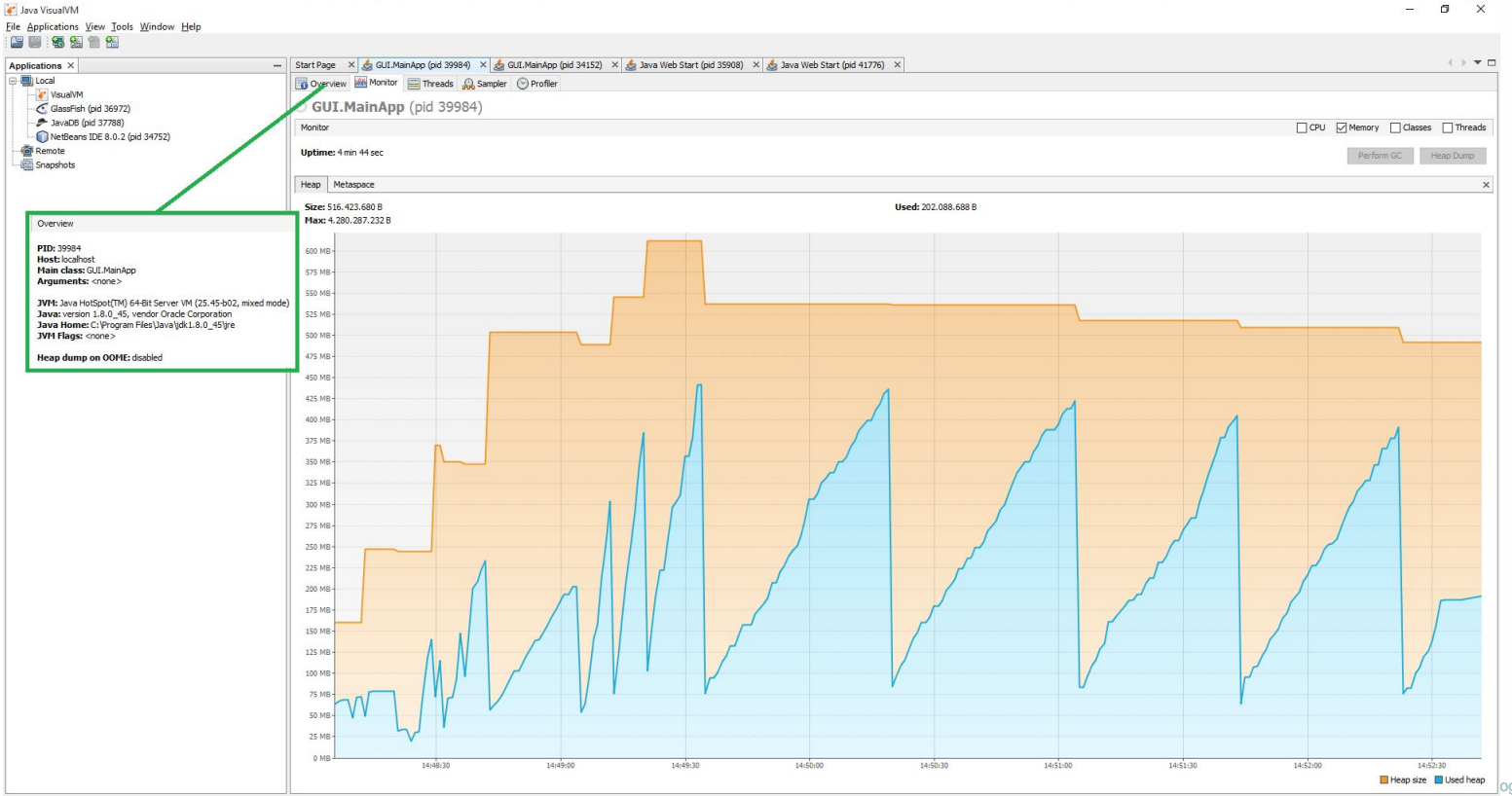
LAB5: JVM MEMORY ISSUES

- Change the Xms and Xmx JVM Heap size to simulate high load

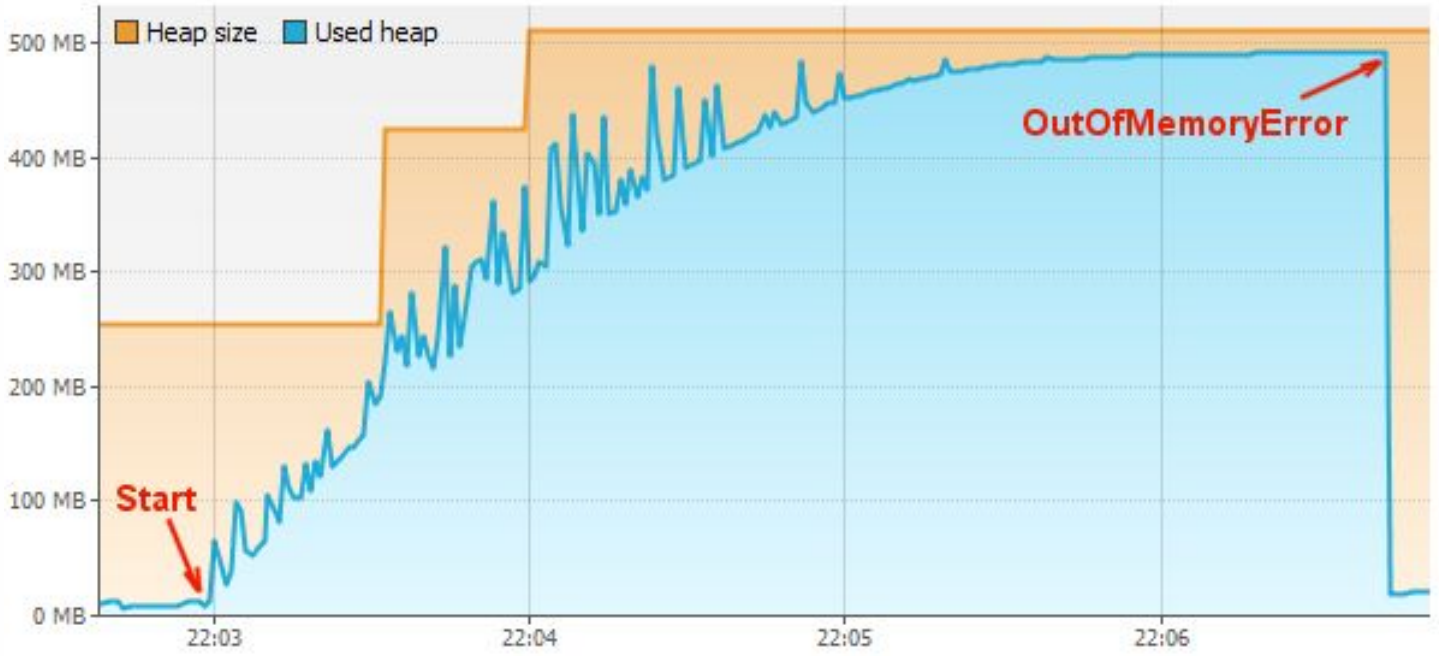
```
helm upgrade artifactory jfrog/artifactory
--version 7.14.3 --set artifactory.javaOpts.xms="256m"
--set artifactory.javaOpts.xmx="512m"
--set artifactory.javaOpts.other="-javaagent:glowroot.jar"
```

- Your Artifactory server will now have a slow response time and the UI will be slow to load
- Lets go to Glowroot and check the JVM state
- See Artifactory JMX Mbeans and add them record them in Glowroot

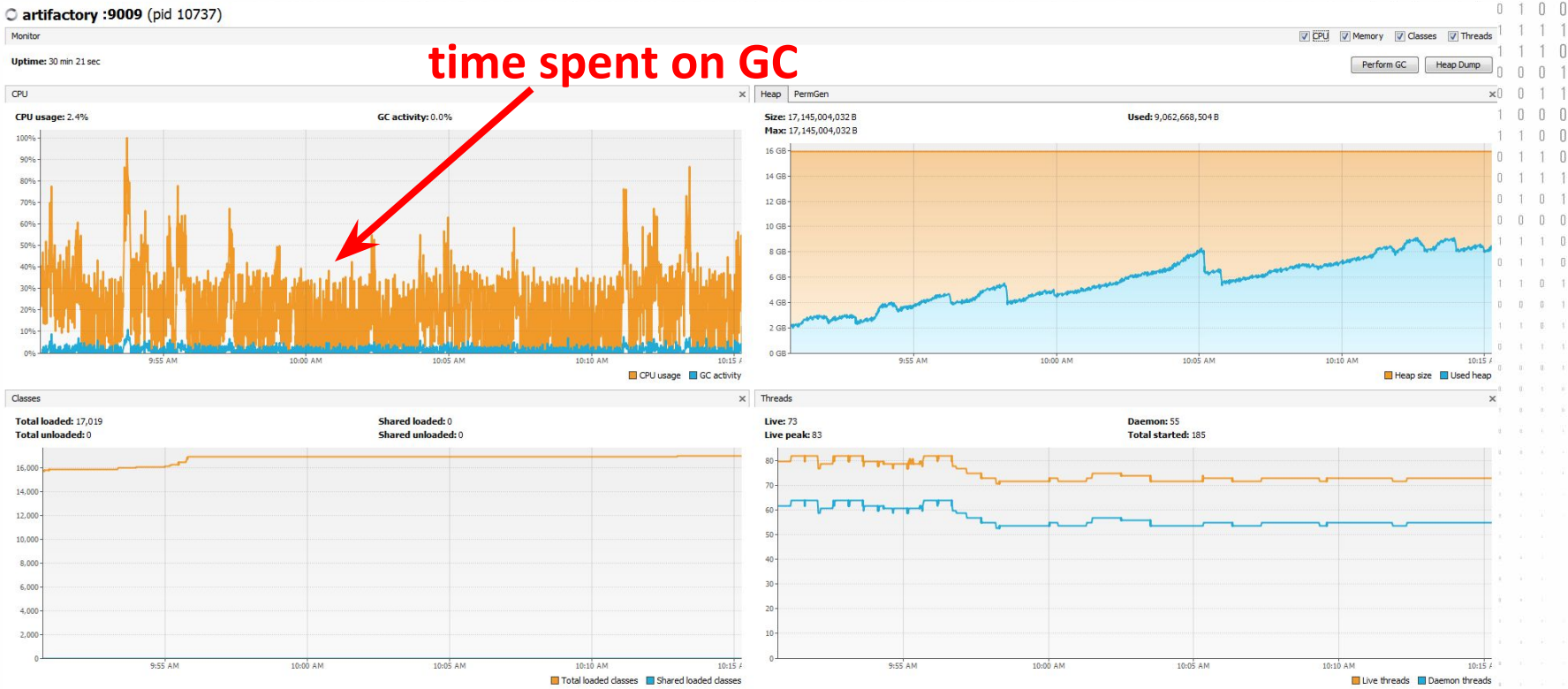
The JVM has a ***heap*** that is the runtime data area from which memory for all class instances and arrays are allocated



JVM - OUT OF MEMORY



EXCESSIVE GARBAGE COLLECTION



HOW DOES IT AFFECT PERFORMANCE?

- Java heap free space is getting lower and lower
- Unable to find a large-enough consecutive section of free memory to be allocated
- At that point, the JVM determines that a garbage collection needs to happen and it notifies the garbage collector

JVM TUNING PARAMETERS

Parameter	Description	Recommendations
-Xmx	Maximum heap size	Should be big enough (e.g. 4 Gb)
-Xms	Initial heap size	Should be ¼ to ½ of Xmx
-Xss	Stack size	Can be small (e.g. 128 Kb)
-XX:+UseG1GC	GC to use	Use Garbage One collector

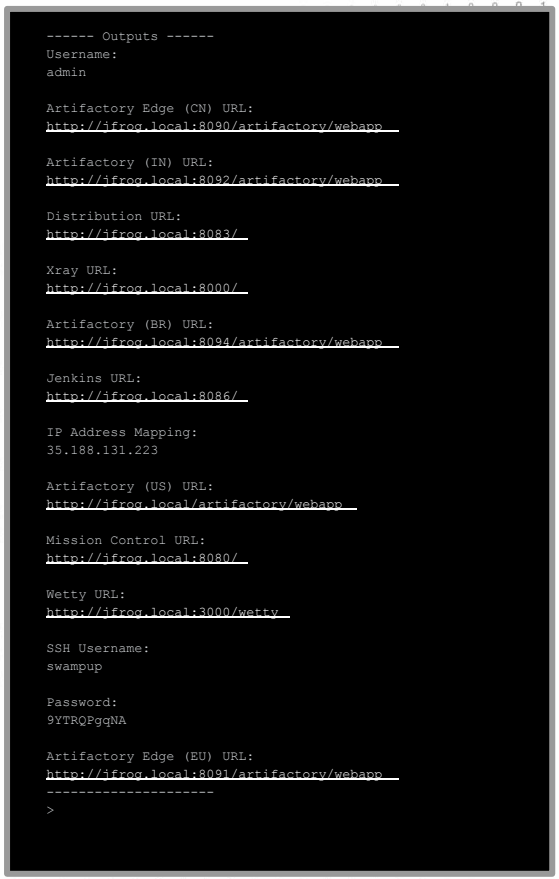
LAB1: UPDATING LOGGING LEVELS

LAB1: UPDATING LOGGING LEVELS

- Set the Artifactory NPM remote URL to “npmm.org” (instead of registry.npmjs.org , the typo is intentional)
- Try to retrieve npm package (using curl) http://x.x.x.x/artifactory/api/npm/npm-remote/express/-/express-2.5.11.tgz
- You should get a 404 response code
- Change to **DEBUG** message level on the Artifactory
- Edit the **logback.xml** file on the Artifactory server configuration to reload automatically

Setup Instructions

- Log into https://jfrog.orbitera.com/c2m/trial/869 and click create an account adding company and contact information
- Click “Create” then “Continue” and accept both agreements
- Click on “Launch Test Drive”
- Wait 9-10 minutes for the JFrog products to start
- You should see the environment log at the bottom
- The instructor will tell you which URL to click on the environment log to start your session
- All logins use admin/<password provided>
- Update your /etc/hosts – jfrog.local to the IP address from the Orbitera environment log (“C:\Windows\System32\Drivers\etc\hosts” on Windows)



Understanding the Environment Details

How are my instances running?

- Each Application is running as a Docker container
- View all containers: “docker ps -a”
- View a specific container log: “docker logs ${CONTAINER_ID} -f”
- Stop an application (for example, restart Artifactory):

  > “docker stop ${CONTAINER_ID}” and to start: “docker start ${CONTAINER_ID}”
- View all aggregated logs from all Artifactory cluster containers:

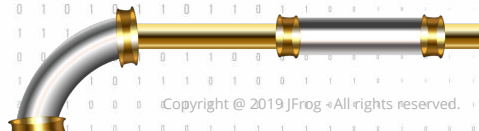
  > “docker-compose -f /root/docker-compose.yml logs -f”

Where is my data?

- All Artifactory instances data: /jfrog/data/artifactory

How can I ssh this environment?

- Wetty ssh URL: http://jfrog.local:3000/wetty
- Username: swampup | pwd: EkJ4NhJS1vjuFcj2fs



ENVIRONMENT DETAILS

Name	UI Port	What	Configuration Location	Container Name
Artifactory (Bangkok)	8094	Artifactory	/jfrog/data/artifactory/s1	artifactory-standalone-1
Artifactory (Cape Town)	8095	Artifactory	/jfrog/data/artifactory/s2	artifactory-standalone-2
Artifactory DR (Denver)	8093	Artifactory	/jfrog/data/artifactory/dr	artifactory-dr
Artifactory HA (Amsterdam)	80	NGINX LB	/jfrog/data/nginx	nginx
Artifactory HA 1	8081	Artifactory HA	/jfrog/data/artifactory/node1	artifactory-node1
Artifactory HA 2	8082	Artifactory HA	/jfrog/data/artifactory/node2	artifactory-node2
Jenkins	8083	Jenkins	/var/lib/jenkins	systemctl stop/start/status jenkins
Mission Control	8080	JFMC	/root/.jfrog/jfmc/jfmc	jfmc_server_1
MySQL	3306	MySQL	/jfrog/data/mysql	mysql
Xray	8000	Xray	/root/.jfrog/xray/xray	xray_xray-server_1