(AUTOMATED) TOMCAT MANAGEMENT

@ladislavGazo

gazo@seges.sk


TOMCAT

STRUCTURE

- **bin** control scripts
- **lib** common libraries
- **webapp** instance context
- **logs** ...

PROBLEM?

NO

for development
(or simple cases)

ADMIN'S PERSPECTIVE

- multiple servers
- multiple instances on a server
- clustered environment
- Tomcat's lack of admin tools
- not willing to pay for **tcServer**

TEMPLATE:

- common logging
- common libraries
- common resource naming (JNDI)

VS.

INSTANCE:

- custom WAR
  - because of different versions
- custom logs
- specific resources (e.g. in JNDI)
- various apps deployed on one server

ENDLESS KILLING

```
/opt/tomcat/bin/shutdown.sh
```

thought it is killed :(

```
ps aux | grep catal
```

get the PID

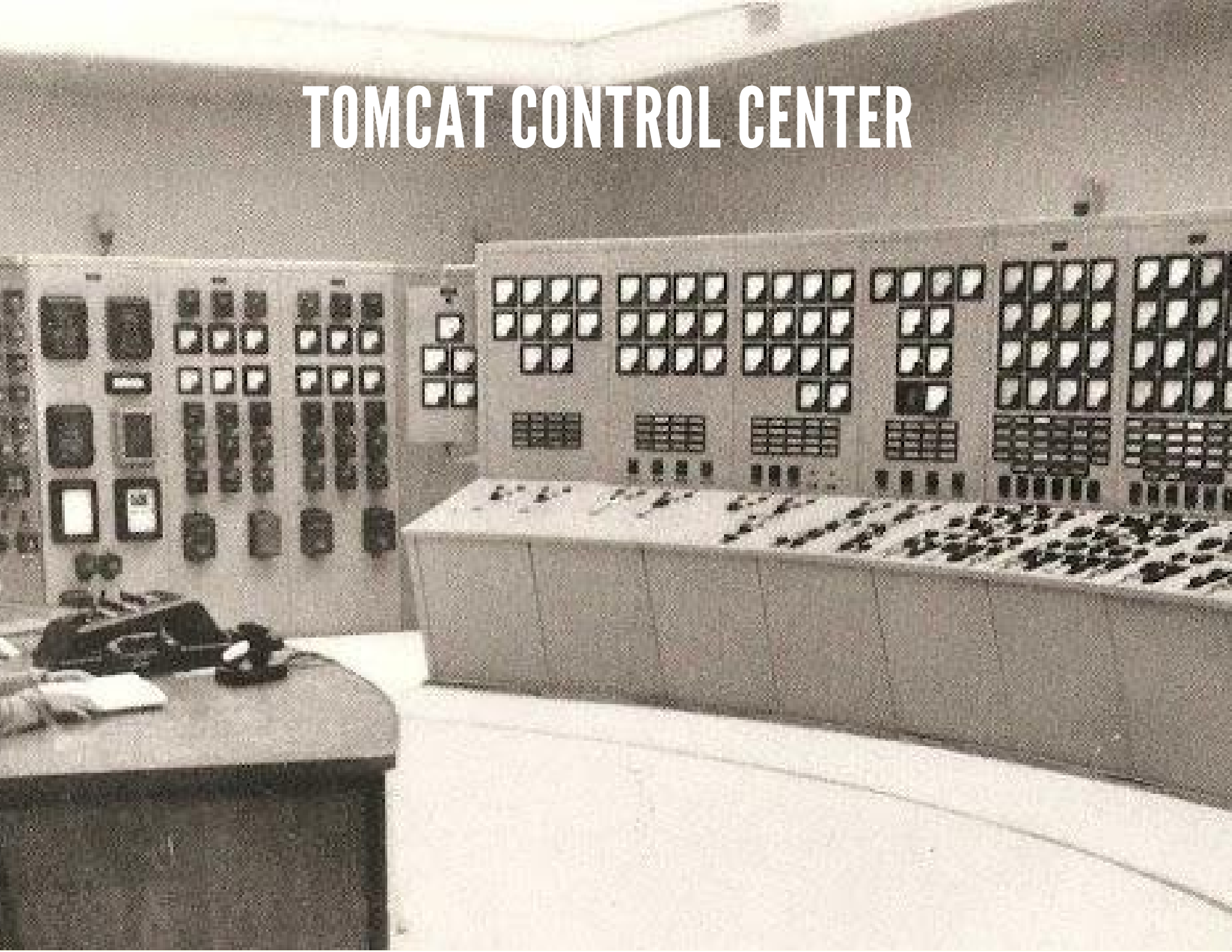
```
kill -9 <da_PID>

cd ../logs
mv * /mnt/backup/logs
cd ../bin
./startup.sh
```

once it is fine but... even in DEV it is boring

... AND WE FACED SUCH ISSUES

TOMCAT CONTROL CENTER



HTTPS://GITHUB.COM/SEGES/TOMCAT-CONTROL-CENTER

WHAT IS IT?

CONCEPT

TEMPLATE

and

(MULTIPLE) INSTANCE(S)

controlled from

SINGLE POINT

[SMALL] SET OF SCRIPTS

start
(start_secured)

stop_force
restart_force
backup_logs_restart_force

status
export_instance_info

control-instances

CENTRAL RESOURCE CONTROL

`TomcatResources.py`

automatically updates server.conf JNDI for each instance
using central CSV files

setenv
resource-db
resource-mail
resource-env
resource-rmi
...

OH GOSH, CSV FILES?

Yeah!

... and yeah! again, I can manage it in Excel :p

PROS

- lightweight and simple
- separate template and instances
- central management

CONS

- so far only on BASH-enabled systems
- no Docker support yet :)
- (not actually a minus) no GUI


DO NOT PUSH!
SAGO 2005


chef

HTTPS://GITHUB.COM/SEGES/CHEF-COOKBOOK-TCC

INSTALLS TCC

PREPARES TEMPLATE(S)

PREPARES INSTANCE(S) ENV

(CAN) MONITOR WITH NEWRELIC


... WISDOM ...


Oh, Yea... This was
definitely...uh.. interesting.

AUTOMATED TCC

```
cd ~/chef

git submodule add https://github.com/seges/chef-cookbook-tcc.git site-cookboo

git submodule add https://github.com/escapestudios-cookbooks/newrelic.git sit
```

CREATE COOKBOOK OR RECIPE

```
knife cookbook create mycookie
```

DEFINE DEFAULT ATTRIBUTES

```
default.tcc.user = "lgazo"
default.tcc.home = "/home/lgazo"
default.tcc.location = "/home/lgazo/opt/tcc"
```

DEFINE TCC TOMCAT TEMPLATE WITH DEPENDENCIES

```
default['tcc']['templates'] = {
 "synapso" => {
   "type" => "tomcat7",
   "libs" => [
     {
       "repo" => "maven",
       "artifact_id" => "org.apache.geronimo.specs:geronimo-jms_1.1_spec:1.1
     },
     {
       "repo" => "maven",
       "artifact_id" => "com.sun.messaging.mq:imq:4.4"
     },
     {
       "repo" => "maven",
       "artifact_id" => "javax.mail:mail:1.4.4"
     },
     {
       "repo" => "maven",
```

FINALLY DEFINE AN INSTANCE

```
default['tcc']['instances'] = {
  "syndev" => {
    "template" => "synapso",
    "user" => "lgazo"
  }
}
```

AND TELL TCC COOKBOOK TO LOOK FOR TEMPLATES IN OUR COOKBOOK

```
default['tcc']['template_cookbooks'] = "mycookie"
```

PREPARE DATABASE IF NEEDED

```
include_recipe "database::postgresql"

postgresql_connection_info = {
  :host     => 'localhost',
  :port     => node['postgresql']['config']['port'],
  :username => 'postgres',
  :password => node['postgresql']['password']['postgres']
}

postgresql_database_user "synapso" do
  connection postgresql_connection_info
  password node.postgresql.password.synapso
  action :create
end

postgresql_database "synapso" do
  connection postgresql_connection_info
```

INCLUDE TCC

```
include_recipe "tcc::default"
include_recipe "tcc::templates"
include_recipe "tcc::instances"<font color="#333333" face="Lato, sans-serif">
</span></font>
```

in a server you can monitor it directly

```
include_recipe "tcc::newrelic"
```

AND NOW THE CONFIGURATION OF INSTANCES

into *templates/default/tcc/templates/synapso/conf*

default environment is named “_default”

- context.xml (in current version it is not Chef environment specific)
- jmx.properties (in current version it is not Chef environment specific)
- resources-
  - c onnector.csv
  - db.csv
  - env.csv
  - mail.csv
  - rmi.csv
  - setenv.csv

TCC MONITORING

JMX

port and access configured in:

- jmx.properties and
- setenv.csv

NEWRELIC

```
include recipe tcc::newrelic
```

QUESTIONS?

@ladislavGazo
gazo@seges.sk


THANK YOU... FOR...
ATTENTION