مستندات **Keyclok**

تنظیمات Wildfly

لازم به ذکر است که بیشتر تنظیمات wildlfy از جمله اینترفیس شبکه، شماره پورت ها و ... دارای پروپرتی متناظر در فایل تنظیمات پروفایل اجرا (مثلا standalone.xml) می باشد. پاس دادن پروپرتی های جاوایی نیز از طریق گزینه –D در خط فرمان صورت می پذیرد. کافی است نگاهی به این فایل بیاندازید . پروپرتی مورد نظر را بیابید.

**بایند کردن به اینترفیس شبکه**

پروپرتی های jboss.bind.address و jboss.bind.address.management اینترفیس های شبکه به منظور بایند شدن اپلیکیشن اصلی keycloak و اپلیکیشن مدیریت wildfly را مشخص می کنند.

مثال:

$ ./standalone.sh -Djboss.bind.address=0.0.0.0 -Djboss.bind.address.management=10.0.1.1

دستور بالا سرویس keycloak را به تمامی اینترفیس های شبکه بایند می کند، اما سرویس مدیریت keycloak را صرفا به اینترفیس 10.0.1.1 بایند می کند. (توصیه می شود به دلایل امنیتی سرویس مدیریت wildfly به صورت عمومی قابل دسترس نباشد)

روش دیگر که برای بایند کردن به اینترفیس شبکه می تواند مورد استفاده قرار گیرد استفاده از گزینه -b می باشد:

$ ./standalone.sh –b 0.0.0.0 -bmanagement 10.0.1.1

مزیت آن در خلاصه نویسی می باشد ولی این روش برای پروپرتی های دیگر قابل استفاده نیست.

بایند کردن به پورت دلخواه

تمامی پورت های مورد استفاده keycloak دارای مقدار پیش فرضی میباشند. در صورت نیاز به استفاده از پورت دیگری از یکی از دو روش زیر می توانید استفاده کنید:

1. استفاده از port-offset
2. ست کردن صریح پورت ها
3. **استفاده از port-offset**

با استفاده از پروپرتی jboss.socket.binding.port-offset می توان تمامی پورت های مورد استفاده توسط keycloak را به عددی دلخواه جمع کرد. مثلا

$ ./standalone.sh -Djboss.socket.binding.port-offset=100

باعث می شود تمامی پورت ها با عدد 100 جمع شوند. بنابراین پورت های مورد استفاده اپلیکیشن سرور 8180, 10090, 7700,8543 و ... خواهند بود.

1. **ست کردن صریح پورت ها**

همانطور که قبلا گفته شد با استفاده پروپرتی ها می توان عددی صریح را به پورت ها در هنگام اجرای wildfly اختصاص داد. برای مشاهده این پروپرتی ها به فایل standalone.xml مراجعه کنید:

$ ./standalone.sh -Djboss.http.port:7070 -Djboss.management.http.port:9991

که پورت keycloak را به 7070 و پورت مدیریت را به 9991 تغییر می دهد.

کلاستر کردن Keycloak

کلاستر کردن Keycloak در محیط های عملیاتی به منظور Highly Available شدن سرویس احراز هویت و جلوگیری از Single Point of Failure شدن سرویس الزامی می باشد. همچنین با کلاستر کردن آن امکان توزیع بار در مواقعی که تعداد درخواست ها زیاد است فراهم می گردد، که باعث ارتقای کیفیت سرویس با کم شدن زمان پاسخگویی می شود.

به منظور کلاستر کردن سرور Keycloak نیاز به اجرای instance های مختلفی از آن (ترجیحا بروی ماشین های مختلف) می باشد. این instance ها را می توان در دو مد Domain و standalone اجرا کرد. در این مستندات ما هر سرور را در مد standalone اجرا می کنیم.

برای اجرای wildlfy در مد standalone دستور زیر را اجرا کنید:

$ ./standalone.sh -c standalone-ha.xml

حال اگر نمونه دیگری از wildfly را با دستور بالا اجرا کنید، در خروجی نمونه اول مشاهده خواهید کرد که آنها تشکیل یک کلاستر داده اند.

دادن نامی دلخواه به هر instance اختیاری می باشد با این وجود قویا توصیه می شود که به هر instance نام مشخصی تخصیص داده شود. مثلا دستور زیر نام node-1 را اختصاص می دهد:

$ ./standalone.sh -c standalone-ha.xml -Djboss.node.name=node-1

instance های مختلف Keycloak نیاز دارند تا به نوعی از وجود یکدیگر اطلاع یابند. و یا به عبارت بهتر تشکیل یک jgroup را بدهند.

در تنظیمات پیش فرض wildfly با استفاده از multicast کردن پروتکل udp از وجود یکدیگر اطلاع یافته و با هم sync می شوند. در این حالت نیاز به هیچ تنظیم اضافه تری نمی باشد.

اما در محیط عملیاتی ممکن است قابلیت multicast شبکه در دسترس نباشد. در این حالت باید از tcpping برای معرفی کردن instance های کلاستر استفاده کرد. بدین منظور در فایل standalone-ha.xml تغییرات هایلات شده را اعمال نمایید.

<subsystem xmlns=**"urn:jboss:domain:jgroups:6.0"**>

<channels default=**"ee"**>

<channel name=**"ee"** stack=**"tcpping"** cluster=**"ejb"**/>

</channels>

<stacks>

<stack name=**"udp"**>

<transport type=**"UDP"** socket-binding=**"jgroups-udp"**/>

<protocol type=**"PING"**/>

<protocol type=**"MERGE3"**/>

<protocol type=**"FD\_SOCK"**/>

<protocol type=**"FD\_ALL"**/>

<protocol type=**"VERIFY\_SUSPECT"**/>

<protocol type=**"pbcast.NAKACK2"**/>

<protocol type=**"UNICAST3"**/>

<protocol type=**"pbcast.STABLE"**/>

<protocol type=**"pbcast.GMS"**/>

<protocol type=**"UFC"**/>

<protocol type=**"MFC"**/>

<protocol type=**"FRAG3"**/>

</stack>

<stack name=**"tcpping"**>

<transport type=**"TCP"** socket-binding=**"jgroups-tcp"**/>

<socket-discovery-protocol type=**"TCPPING"** socket-bindings=**"jgroups-host-a jgroups-host-b"**/>

<protocol type=**"MERGE3"**/>

<protocol type=**"FD\_SOCK"**/>

<protocol type=**"FD\_ALL"**/>

<protocol type=**"VERIFY\_SUSPECT"**/>

<protocol type=**"pbcast.NAKACK2"**/>

<protocol type=**"UNICAST3"**/>

<protocol type=**"pbcast.STABLE"**/>

<protocol type=**"pbcast.GMS"**/>

<protocol type=**"MFC"**/>

<protocol type=**"FRAG2"**/>

</stack>

</stacks>

</subsystem>

حال در بخش socket-binding-group آدرس هاست را وارد نمایید:

<outbound-socket-binding name=**"jgroups-host-a"**>

<remote-destination host=**"10.0.1.1"** port=**"7600"**/>

</outbound-socket-binding>

<outbound-socket-binding name=**"jgroups-host-b"**>

<remote-destination host=**"10.0.1.2"** port=**"7600"**/>

</outbound-socket-binding>

ایراد tcpping این است که نیاز داریم آدرس IP و پورت هاست را به صورت استاتیک در فایل تنظیمات پروژه وارد نماییم.

استفاده از Load Balancer

به منظور لود بالانس کردن بار سرورها نیاز به یک لودبالانسر می باشد.

1. Apache HTTP Server + mod\_cluster
2. NGINX به عنوان Reverse Proxy

در این مستندات نحوه استفاده از NGINX در نقش Reverse Proxy شرح داده خواهد شد.

تنظیمات NGINX برای لودبالانس کردن Keycloak

1. تنظیم NGINX به عنوان Reverse Proxy برای لود بالانس کردن سرویس Keycloak

در دایرکتوری /etc/nginx/conf.d/ فایل با نامی دلخواه، مثلا jboss.conf بسازید و خطوط زیر را در آن کپی کنید:

proxy\_cache\_path /tmp/NGINX\_cache/ keys\_zone=backcache:10m;

map $http\_upgrade $connection\_upgrade {

default upgrade;

'' close;

}

upstream jboss {

# Use IP Hash for session persistence

ip\_hash;

# List of JBoss Application Servers

server 192.168.100.117:8080;

server 192.168.100.118:8080;

}

server {

listen 192.168.100.118:2020;

server\_name tjb-sso.com;

location / {

proxy\_pass http://jboss;

proxy\_set\_header Host $http\_host;

proxy\_set\_header X-REAL-IP $remote\_addr;

proxy\_set\_header X-FORWARDED-FOR $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-FORWARDED-PROTO $scheme;

}

}

نکته حایز اهمیت در تنظیمات بالا ست کردن برخی هدرها از سمت NGINX است. NGINX IP واقعی کلاینت را در این هدرها ست کرده و آن را به Keycloak ارسال می نماید.

همچنین نیاز دارید به Wildfly بگویید که IP واقعی کلاینت را از هدر مربوطه بازیابی کند، بدین منظور در فایل تنظیمات wildfly، بسته به پروفایل مورد استفاده مثلا standalone-ha.xml، تغییر زیر را اعمال کنید:

<subsystem xmlns="urn:jboss:domain:undertow:6.0">

<buffer-cache name="default"/>

<server name="default-server">

<ajp-listener name="ajp" socket-binding="ajp"/>

<http-listener name="default" socket-binding="http" redirect-socket="https"

proxy-address-forwarding="true"/>

...

</server>

...

</subsystem>

اکنون نیاز دارید که سرویس NGINX را راه اندازی مجدد کنید:

# systemctl restart nginx

برای اطلاعات بیشتر به آدرس زیر مراجعه کنید:

<https://docs.nginx.com/nginx/deployment-guides/jboss-load-balancing-nginx-plus/>

1. خطا در بایند کردن NGINX به پورت دلخواه

ممکن است هنگام استارت کردن NGINX خطای "Permission denied" را دریافت کنید. در صورتی که SELinux فعال باشد، ممکن است خطا از سمت SELinux باشد. برای مشاهده پورت هایی که اجازه بایند شدن به آنها وجود دارد، دستور زیر را وارد کنید:

# Semanage port –l | grep http\_port\_t

http\_port\_t tcp 80, 81, 443, 448, 8008, 8009, 8443, 9000

اگر پورت مورد نظر را در لیست بالا مشاهده نمی کنید به معنای آنست که SELinux اجازه بایند کردن به آن پورت را نمی دهد، در اینصورت با وارد کردن دستور زیر پورت مورد نظر را در لیست پورت های مجاز قرار دهید:

# semanage port -a -t http\_port\_t -p tcp 2020

حال اگر مجددا دستور نمایش لیست پورت های مجاز را وارد کنید باید پورت مورد نظر را در لیست مشاهده کنید:

# Semanage port –l | grep http\_port\_t

http\_port\_t tcp 80, 81, 443, 448, 2020, 8008, 8009, 8443, 9000

1. خطا در forward کردن درخواست توسط NGINX

در صورتی که NGINXبه عنوان Reverse Proxy سرویس دیگری تنظیم شده باشد، در صورت فعال بودن SELinux ممکن است NGINX امکان forward کردن درخواست ها را نداشته باشد. در این صورت شما خطای 502 Bad Gateway را دریافت خواهید کرد.

برای رفع این خطا دستور زیر را وارد کنید:

# setsebool -P httpd\_can\_network\_connect true