

```
# Determines HA Status, creates variable for policy hashes, then loops through it on Active devices to
return the number of signatures that are ready for enforcement on each policy
# SETUP1: tmsh modify /sys outbound-smtp mailhub smtp.example.com
# SETUP2: tmsh modify /sys outbound-smtp from-line-override enabled
```

```
bash
cd /var/tmp/
```

```
# Static Variables
CREDS=admin
```

```
# Exits script for standby devices
haStatus=$(tmsh show /cm failover-status | grep Status)
if [[ $haStatus == *"STANDBY"* ]]; then
    exit
fi
```

```
# Build HTML Page
cat > /var/tmp/ready-suggestions.html
echo '<h1>ASM Learning Suggestions Ready for Review Per Policy</h1>' >> /var/tmp/ready-
suggestions.html
echo '<table cellpadding="3" bgcolor="#000000"><tr bgcolor="#ffffff"><th>Policy
Name</th><th>Number of Suggestions Ready for Review</th><th>Blocking
Mode</th><th>Microservices</th></tr>' >> /var/tmp/ready-suggestions.html
```

```
# Creates variable with list of policy hashes, then prints variable contents to txt file
#FILENAME=$(curl -kvu $CREDS http://localhost/mgmt/tm/asm/policies | jq -r '.items[].id')
FILENAME=$(curl -kvu $CREDS http://localhost/mgmt/tm/asm/policies | jq -r '.items[] | .id')
printf "$FILENAME\n" > /var/tmp/policy-hashes.txt
```

```
FILENAME="policy-hashes.txt"
LINES=$(cat $FILENAME)
```

```
# Assigns ready signatures per policy to OUTPUT variable, then does a wc -l to count the number of lines
in that variable, and writes the result to the HTML file
for LINE in $LINES
do
    OUTPUT=$(curl -kvu $CREDS -X GET
"https://localhost/mgmt/tm/asm/policies/$LINE/suggestions?\$filter=status+eq+%27pending%27+and+
score+eq+100" -u $CREDS -k -v -H "Content-Type: application/json" | jq -r '.totalItems')
    BLOCKINGMODE=$(curl -kvu $CREDS -X GET "https://localhost/mgmt/tm/asm/policies/$LINE" -u
$CREDS -k -v -H "Content-Type: application/json" | jq -r '.enforcementMode')
    MICROSERVICES=$(curl -kvu $CREDS -X GET
"https://localhost/mgmt/tm/asm/policies/$LINE/microservices" -u $CREDS -k -v -H "Content-Type:
application/json" | jq -r '.totalItems')
    POLICYNAMES=$(curl -kvu $CREDS http://localhost/mgmt/tm/asm/policies/$LINE | jq -r '.name')
```

```
echo "<tr
bgcolor="#ffffff"><td>$POLICYNAMES</td><td>$OUTPUT</td><td>$BLOCKINGMODE</td><td>$MICRO
SERVICES</td></tr>" >> /var/tmp/ready-suggestions.html
done
```

```
echo '</table>' >> /var/tmp/ready-suggestions.html
```

```
# Dynamically sets the email source address as the F5 hostname, then emails the content of the ready-
suggestions.html file
```

```
hostname=$(tmsh list sys global-settings hostname | grep hostname)
```

```
hostname=${hostname//hostname/}
```

```
value=$(cat /var/tmp/ready-suggestions.html)
```

```
(echo To: email@example.com
```

```
echo From: $hostname
```

```
echo "Content-Type: text/html; "
```

```
echo Subject: Learning Suggestions Ready for Review Per Policy
```

```
echo
```

```
echo "$value" ) | sendmail -t
```