느린 APIC GUI 문제 해결

목차

소개

이 문서에서는 느린 APIC GUI 환경을 트러블슈팅하는 일반적인 방법론에 대해 설명합니다.

빠른 시작

느린 APIC GUI 문제는 스크립트, 통합 또는 애플리케이션에서 발생하는 API 요청 비율이 높기 때문에 발생하는 경우가 많습니다. APIC의 access.log는 처리된 각 API 요청을 기록합니다. APIC의 access.log는 Github Datacenter 그룹 aci-tac-scripts 프로젝트 내의 Access Log Analyzer 스크립트를 사용하여 신속하게 분석할 수 있습니다.

배경 정보

웹 서버로서의 APIC - NGINX

NGINX는 각 APIC에서 사용 가능한 API 엔드포인트를 담당하는 DME입니다. NGINX가 다운되면

API 요청을 처리할 수 없습니다. NGINX가 혼잡할 경우 API가 혼잡합니다. 각 APIC는 자체 NGINX 프로세스를 실행하므로, 공격적인 쿼리 발송자의 표적이 된 APIC만 NGINX 문제를 일으킬 수 있습니다.

APIC UI는 여러 API 요청을 수행하여 각 페이지를 채웁니다. 마찬가지로, 모든 APIC 'show' 명령(NXOS Style CLI)은 여러 API 요청을 수행하고 응답을 처리한 다음 사용자에게 제공하는 python 스크립트의 래퍼입니다.

관련 로그

로그 파일 이름	위치	어느 기술 지원 부서에 있습니까	의견
access.log	/var/log/dme/log	APIC 3of3	ACI에 구애받지 않으며, API 요청당 라인 1개 제공
error.log	/var/log/dme/log	APIC 3of3	ACI에 구애받지 않음, nginx 오류 표시(조절 포함)
nginx.bin.log	/var/log/dme/log	APIC 3of3	ACI별, DME 트랜잭션 기록
nginx.bin.warnplus.log	/var/log/dme/log	APIC 3of3	ACI Specific(ACI 특정)에는 경고 이상의 심각도가 있는 로그가 포함되어 있습니다.

방법론

초기 트리거 격리

어떤 영향을 받습니까?

- 영향을 받는 APIC는 무엇입니까? APIC 하나, 여러 개 또는 모두?
- UI, CLI 명령 또는 둘 다를 통해 느림이 나타나는 곳은?
- 어떤 특정 UI 페이지 또는 명령이 느립니까?

느림은 어떻게 느껴지나요?

- 단일 사용자가 여러 브라우저에서 볼 수 있습니까?
- 여러 사용자가 느리게 보고합니까, 아니면 단일/하위 집합 사용자만 보고합니까?
- 영향을 받는 사용자는 브라우저에서 APIC으로 가는 지리적 위치나 네트워크 경로가 비슷합니까?

느림이 처음 알려진 때는 언제였습니까?

- 최근 ACI 통합 또는 스크립트가 추가되었습니까?
- 최근에 브라우저 확장이 활성화되었습니까?
- 최근 ACI 컨피그레이션이 변경되었습니까?

NGINX 사용 및 상태 확인

Access.log 항목 형식

access.log는 NGINX의 기능이므로 APIC에 구애받지 않습니다. 각 행은 APIC에서 수신한 1개의 HTTP 요청을 나타냅니다. 이 로그를 참조하여 APIC의 NGINX 사용을 파악합니다.

ACI 버전 5.2+의 기본 access.log 형식:

```
log_format proxy_ip '$remote_addr ($http_x_real_ip) - $remote_user [$time_local]'
                    '"$request" $status $body_bytes_sent '
                    '"$http_referer" "$http_user_agent"';
```

이 행은 moquery -c fvTenant를 수행할 때 access.log 항목을 나타냅니다.

```
127.0.0.1 (-) - - [07/Apr/2022:20:10:59 +0000]"GET /api/class/fvTenant.xml HTTP/1.1" 200 15863 "-" "Pyt
```

access.log 항목의 예를 log_format에 매핑합니다.

log_format 필드	예제의 내용	의견
$remote_addr	127.0.0.1	이 요청을 보낸 호스트의 IP
$http_x_real_ip	-	프록시가 사용 중인 경우 마지막 요청자의 IP
$remote_user	-	일반적으로 사용되지 않습니다. 요청을 수행하기 위해 로그인한 사용자를 추적하려면 nginx.bin.log를 선택합니다
$time_local	07/4/2022:20:10:59 +000	요청이 처리되었을 때

$request(요청)	/api/class/fvTenant.xml HTTP/1.1 다운로드	Http 메서드(GET, POST, DELETE) 및 URI
$status	200	HTTP 응답 상태 코드
$body_bytes_sent	1586	응답 페이로드 크기
$http_referer	-	-
$http_user_agent	피톤우를리브	요청을 보낸 클라이언트 유형

Access.log 동작

대량의 시간에 걸친 고속 요청 버스트:

- 초당 15개 이상의 요청이 지속적으로 급증할 경우 UI 속도가 느려질 수 있습니다.
- 어떤 호스트가 쿼리를 담당하는지 파악
- 쿼리 소스를 줄이거나 비활성화하여 APIC 응답 시간이 개선되는지 확인합니다.

일관된 4xx 또는 5xx 응답:

- 발견된 경우 nginx.bin.log에서 오류 메시지를 식별합니다

NGINX 리소스 사용량 확인

NGINX CPU 및 메모리 사용량은 APIC의 top 명령으로 확인할 수 있습니다.

<#root>

```
top - 13:19:47 up 29 days, 2:08, 11 users, load average: 12.24, 11.79, 12.72
Tasks: 785 total, 1 running, 383 sleeping, 0 stopped, 0 zombie
%Cpu(s): 3.5 us, 2.0 sy, 0.0 ni, 94.2 id, 0.1 wa, 0.0 hi, 0.1 si, 0.0 st
KiB Mem : 13141363+total, 50360320 free, 31109680 used, 49943636 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 98279904 avail Mem

PID   USER PR NI VIRT    RES   SHR   S %CPU %MEM TIME+     COMMAND
21495 root 20 0  4393916 3.5g 217624 S
```

2.6

2.8 759:05.78

nginx.bin

높은 NGINX 리소스 사용량은 처리된 요청의 높은 비율과 직접 관련이 있을 수 있습니다.

코어 확인

NGINX 충돌은 느린 APIC GUI 문제에 대해 일반적이지 않습니다. 그러나 NGINX 코어가 발견되면 TAC SR에 연결하여 분석하십시오. 코어를 확인하는 단계는 ACI Techsupport 가이드를 참조하십시오.

클라이언트 대 서버 레이턴시 확인

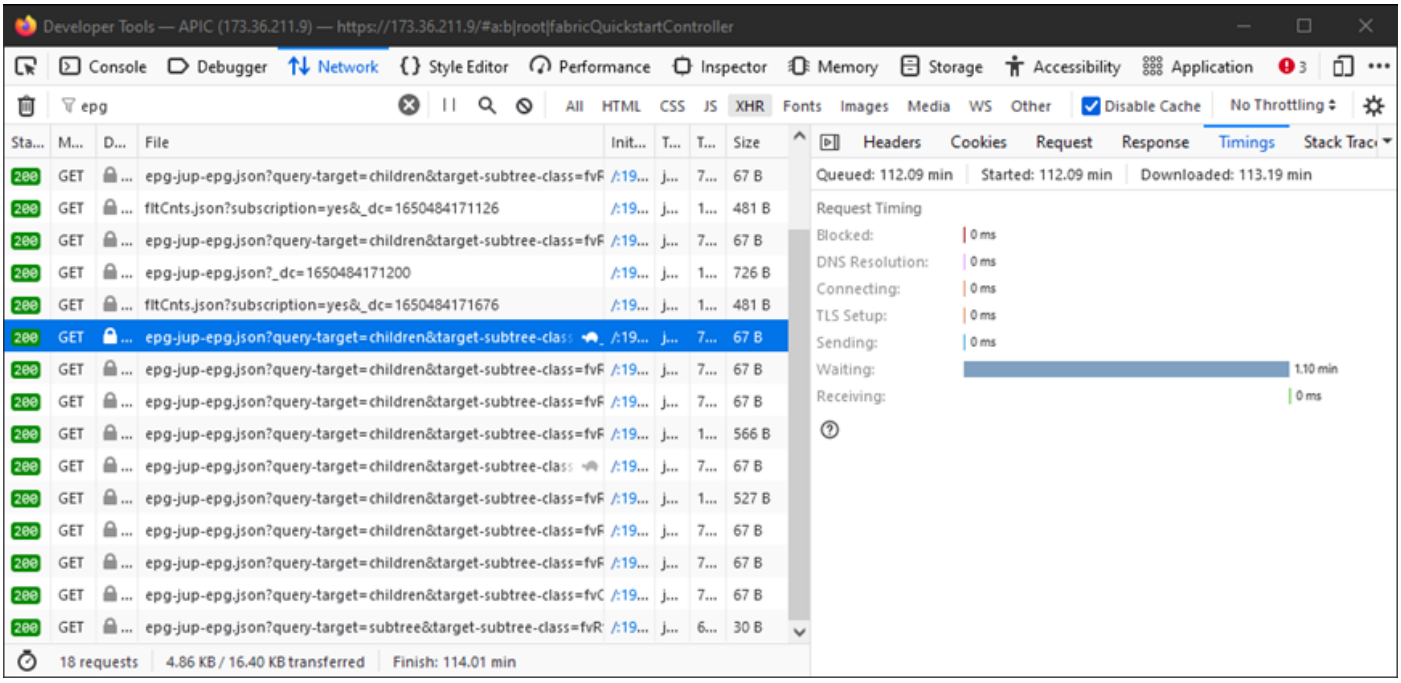
빠른 요청을 찾을 수 없지만 사용자가 계속 UI 느림을 나타내는 경우 클라이언트(브라우저) 대 서버(APIC) 레이턴시가 문제가 될 수 있습니다.

이러한 시나리오에서는 브라우저에서 APIC까지의 데이터 경로(지리적 거리, VPN 등)를 검증합니다. 가능한 경우, 격리할 APIC와 동일한 지리적 지역 또는 데이터 센터에 위치한 점프 서버에서 액세스를 구축하고 테스트합니다. 다른 사용자가 유사한 레이턴시를 나타내는지 확인합니다.

브라우저 개발 도구 네트워크 탭

모든 브라우저에는 일반적으로 네트워크 탭 내에서 브라우저 개발 툴킷을 통해 HTTP 요청 및 응답을 검증하는 기능이 있습니다.

이 도구는 이미지에 표시된 대로 브라우저 소스 요청의 각 단계에 소요되는 시간을 검증하는 데 사용할 수 있습니다.



APIC가 응답하기 위해 1.1분 동안 대기하는 브라우저의 예

특정 UI 페이지에 대한 개선 사항

정책 그룹 페이지:

Cisco 버그 ID CSCvx14621 - 패브릭 탭의 IPG 정책에서 APIC GUI가 느리게 로드됩니다.

Inventory(인벤토리) 페이지의 인터페이스:

Cisco 버그 ID CSCvx90048 - "Layer 1 물리적 인터페이스 컨피그레이션" Operational(운영) 탭의 초기 로드가 길고 '고정'을 유도합니다.

General Recommendations for Client(클라이언트 > 서버 레이턴시에 대한 일반 권장 사항)

Firefox와 같은 특정 브라우저는 기본적으로 호스트당 더 많은 웹 연결을 허용합니다.

- 사용되는 브라우저 버전에서 이 설정을 구성할 수 있는지 확인
- 이는 Policy Group(정책 그룹) 페이지와 같은 다중 쿼리 페이지에 더 중요합니다

VPN 및 APIC과의 거리는 클라이언트 브라우저 요청 및 APIC 응답 이동 시간을 감안할 때 전반적인 UI 느려짐을 증가시킵니다. APIC에서 지리적으로 로컬인 점프 박스는 브라우저를 APIC 이동 시간으로 대폭 단축합니다.

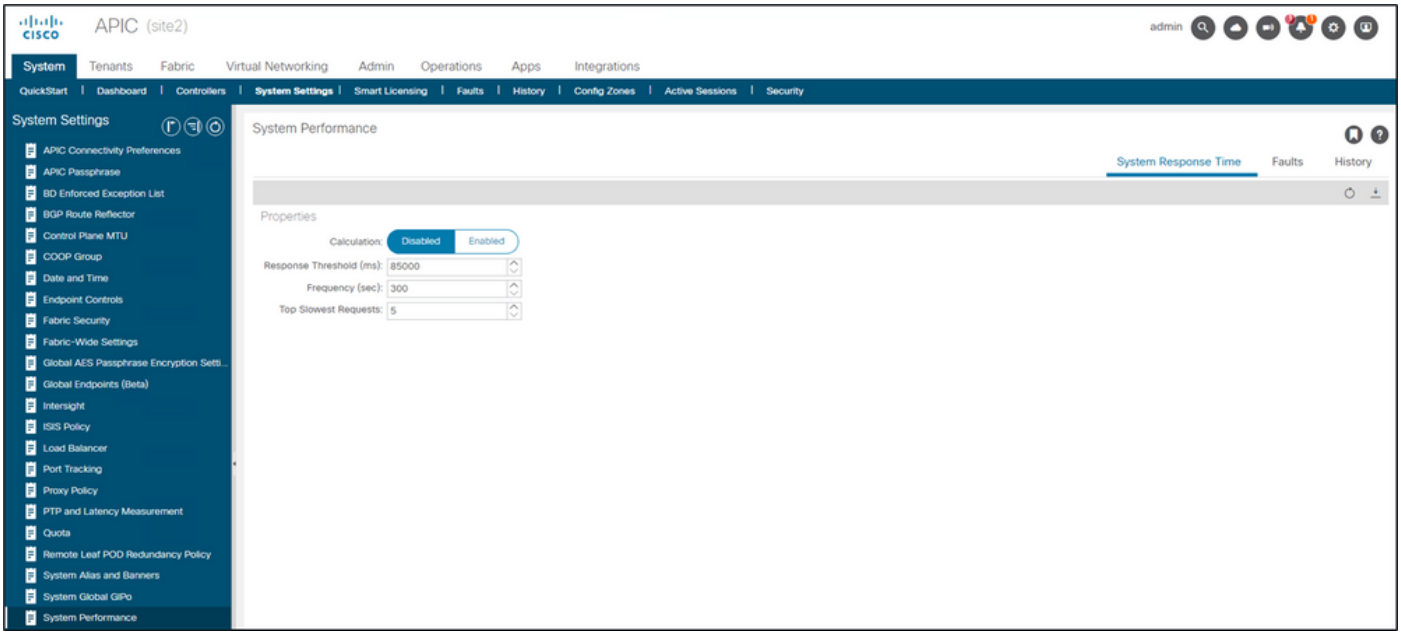
Long-Web 요청 확인

웹 서버(APIC의 NGINX)가 많은 양의 Long-Web 요청을 처리하는 경우, 이는 병렬로 수신된 다른 요청의 성능에 영향을 줄 수 있습니다.

특히 APIC와 같이 데이터베이스가 분산된 시스템에서는 이러한 현상이 두드러집니다. 단일 API 요청에는 패브릭의 다른 노드로 전송되는 추가 요청 및 조회가 필요할 수 있으며, 이로 인해 응답 시간이 더 길어질 수 있습니다. 짧은 시간 내에 이러한 Long-Web Requests를 버스트하면 필요한 리소스의 양이 늘어나 응답 시간이 예기치 않게 더 길어질 수 있습니다. 또한 수신된 요청은 시간 초과(90초)되어 사용자 관점에서 예기치 않은 시스템 동작이 발생할 수 있습니다.

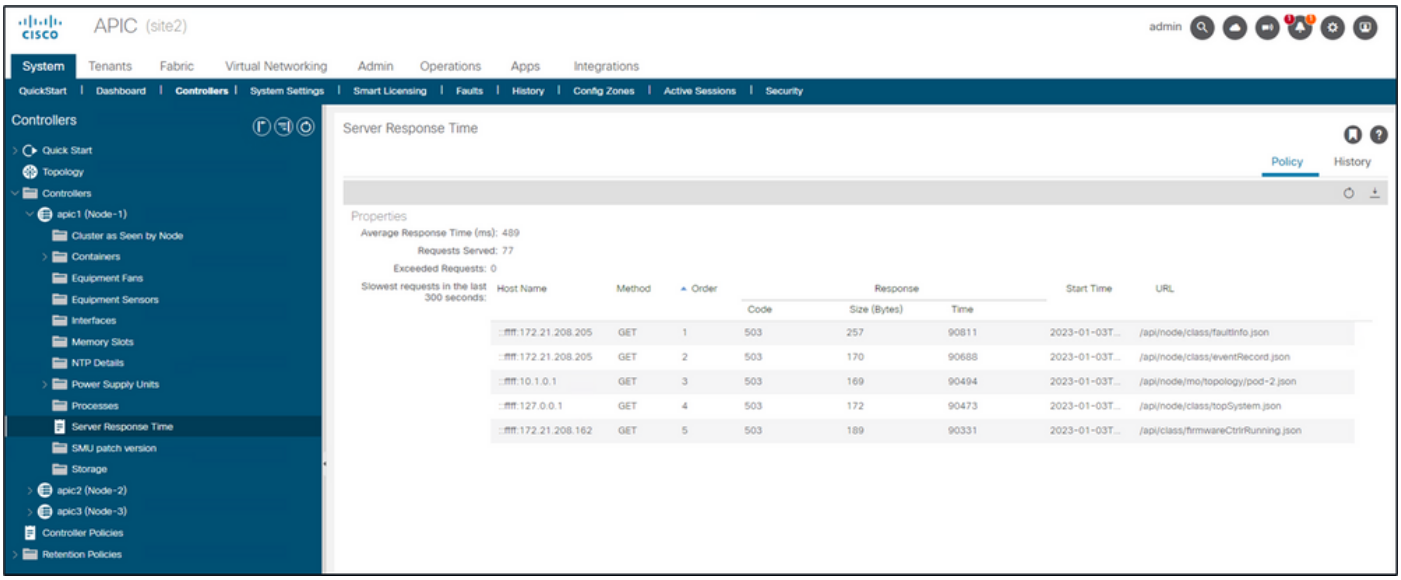
시스템 응답 시간 - 서버 응답 시간에 대한 계산 사용

4.2(1)+에서 사용자는 처리 시간이 소요된 API 요청을 추적하고 강조 표시하는 "시스템 성능 계산"을 활성화할 수 있습니다.



System - System Settings - System Performance에서 계산을 활성화할 수 있습니다.

"계산"이 활성화되면 사용자는 컨트롤러 아래의 특정 APIC로 이동하여 최근 300초 내에 가장 느린 API 요청을 볼 수 있습니다.



시스템 - 컨트롤러 - 컨트롤러 폴더 - APIC x - 서버 응답 시간

APIC API 사용 고려 사항

스크립트가 Nginx를 손상시키지 않도록 하는 일반적인 조언

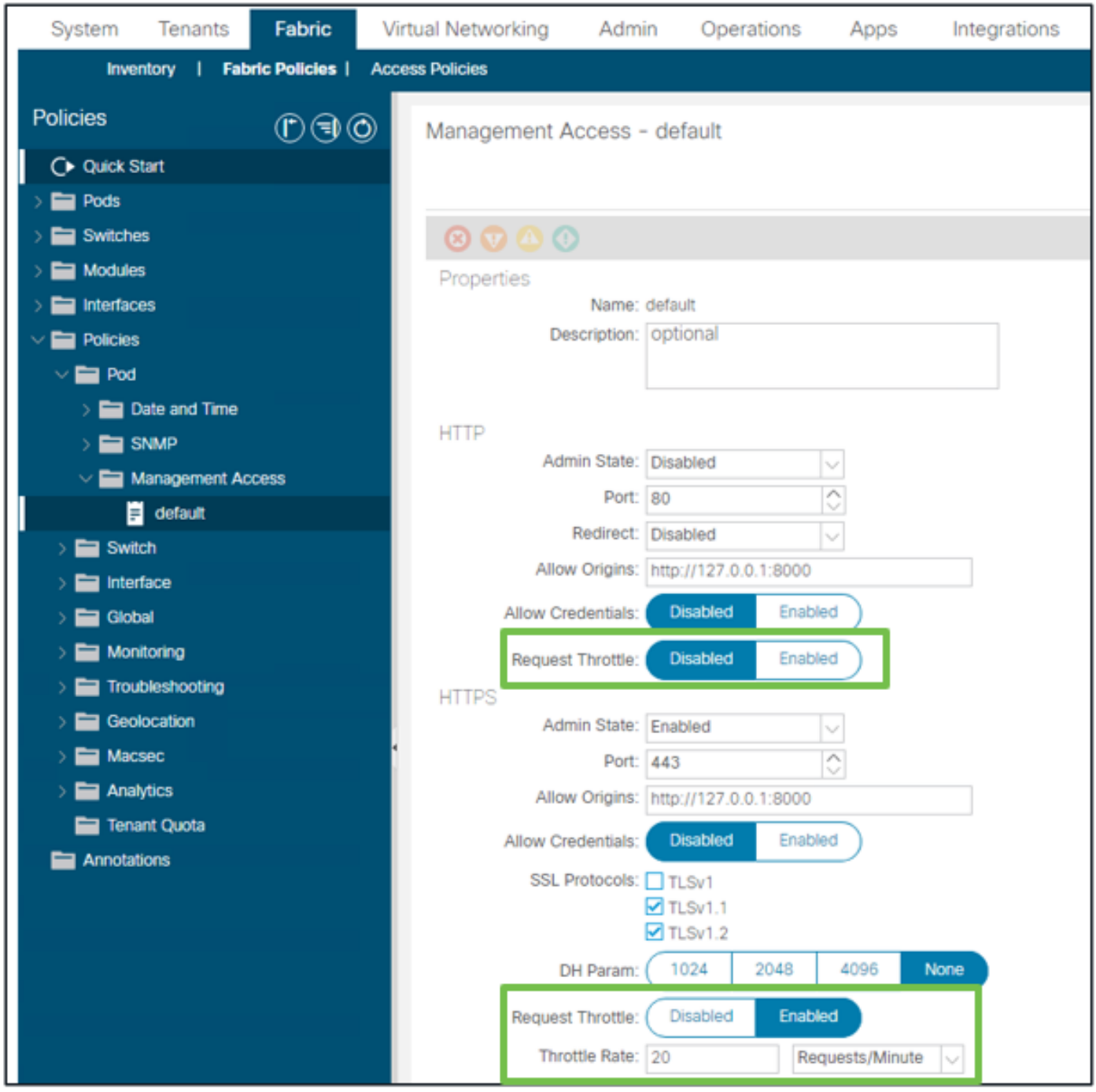
- 각 APIC는 자체 NGINX DME를 실행합니다.
  - APIC 1의 NGINX만 APIC 1에 대한 요청을 처리합니다. APIC 2와 3의 NGINX는 이러한 요청을 처리하지 않습니다.
- 일반적으로 NGINX는 장기간에 걸쳐 초당 15개 이상의 API 요청을 통해 서비스 품질을 저하시킵니다.
  - 발견된 경우 요청의 적극성을 줄입니다.
  - 요청 호스트를 수정할 수 없는 경우 APIC에서 NGINX Rate Limits를 고려하십시오.

스크립트 비효율성 해결

- 각 API 요청 전에 로그인/로그아웃하지 마십시오.
  - 한 로그인 세션의 기본 시간 제한은 10분입니다. 이 동일한 세션을 여러 요청에 사용할 수 있으며 유효 기간을 연장하기 위해 새로 고칠 수 있습니다.
  - Cisco APIC REST API 컨피그레이션 가이드 - REST API 액세스 - API 세션 인증 및 유지 관리를 참조하십시오.
- 스크립트에서 상위 항목을 공유하는 여러 DN을 쿼리하는 경우 쿼리 필터를 사용하여 쿼리를 단일 논리 상위 쿼리로 축소하지 않습니다.
  - Cisco APIC REST API 컨피그레이션 가이드 - REST API 쿼리 작성 - 쿼리 범위 필터 적용을 참조하십시오.
- 객체 또는 객체 클래스의 업데이트가 필요한 경우 빠른 API 요청 대신 웹소켓 서브스크립션을 고려하십시오.

NGINX 요청 제한

4.2(1)+에서 사용할 수 있으며, 사용자는 HTTP 및 HTTPS에 대해 요청 제한을 독립적으로 활성화할 수 있습니다.



패브릭 - 패브릭 정책 - 정책 폴더 - 관리 액세스 폴더 - 기본값

활성화된 경우:

- 구성 파일 변경 사항을 적용하기 위해 NGINX가 다시 시작됩니다.
  - 새 영역 httpsClientTagZone이 nginx 구성에 기록됩니다
- 스로틀 속도는 Requests per Minute(r/m) 또는 Requests per Second(r/s)로 설정할 수 있습니다.
- 요청 스로틀은 NGINX에 포함된 속도 제한 구현에 의존함

- /api/URI에 대한 API 요청은 사용자 정의 Throttle Rate + burst= (Throttle Rate x 2) + nodelay를 사용합니다
  - /api/aaaLogin 및/api/aaaRefresh에는 2r/s + burst=4 + nodelay에서 속도 제한을 설정할 수 없는 스로틀(영역 aaaApiHttps)이 있습니다
- 요청 스로틀은 클라이언트 IP 주소별로 추적됩니다.
- APIC 셀프 IP(UI + CLI)에서 제공된 API 요청이 스로틀을 우회함
- 사용자 정의 스로틀 속도 + 버스트 임계값을 초과하는 모든 클라이언트 IP 주소는 APIC에서 503 응답을 수신합니다
- 이러한 503은 액세스 로그 내에서 상관관계가 있을 수 있습니다
- error.log에는 제한이 활성화된 시점(영역 httpsClientTagZone) 및 어떤 클라이언트 호스트에 대해 표시되는지 나타내는 항목이 있습니다

<#root>

apic#

less /var/log/dme/log/error.log

...
2023/04/17 20:19:14 [error] ...

limiting requests

, excess: 40.292 by zone "

httpsClientTagZone

", client: h.o.s.t, ... request: "GET /api/class/...", host: "a.p.i.c"
2023/04/17 20:19:14 [error] ...

limiting requests

, excess: 40.292 by zone "

httpsClientTagZone

", client: h.o.s.t, ... request: "GET /api/node/...", host: "a.p.i.c"

일반적으로 Request Throttle은 쿼리 집약적 클라이언트에 의해 유발되는 DDOS 유사 증상으로부터 서버(APIC)를 보호하는 역할만 합니다. 애플리케이션/스크립트 논리의 최종 솔루션에 대한 request-aggressive Client를 이해하고 격리합니다.

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.