

سوال 1-نسخه های آپدیت شده ان به ترتیب د به همین خاطر دوتا دوتا مقایسه شدن

http1.0	http1.1	
تک درخواست/جواب	چندین درخواست/جواب	به ازای هر کانکشن
اختیاری	الزامی	هدر هاست
کمتر	بیشتر	امنیت
محدود	هدر شرطی	کشینگ
ندارد	پروکسی، احراز هویت	اعتبارسنجی
ندارد	دارد	رمزنگاری انتقال قطعه ای
ندارد	دارد	هدر کانکشن
16 تا	24 تا حالت های بیشتری دارد	کد استیت

http 1.1	http2	
آیدی جفت درخواست/جواب(همزمانی)	دارد در هر فریم(مالتی پلکسینگ)	ندارد
سرعت لود	بیشتر (به خاطر فشرده سازی هدر)~ 8 برابر	کمتر
فرمت درخواست/جواب	فریم باینری (استریم)	تکست

http2	http3	
پروتکل لایه کاربرد http	تغییر در پروتکل لایه انتقال	هندل کردن استریم
TCP & TLS	UDP(QUIC)	پروتکل انتقال
کمتر	بیشتر	سرعت لود

به خاطر اینکه UDP کنترل ازدحام ندارد و اطمینان رو رسیدن پکت نمیده فقط دوباره ارسال میکنه این باعث میشه درخواست ها منتظر همدیگه بخاطر ازدحام نمونن و سریعتر ارسال شوند.

سوال 2-

الف) در stateless: لزوم نداره که سرور اطلاعات سرور یا جزئیات سیشن ها رو حفظ کنه به همین خاطر ساده است پکت ها مستقل اند چون نیازی نیست به هم دیگه به عنوان مرجع اشاره کنند ارتباط نزدیکی وابستگی برای دسترسی به اطلاعات بین کلاینت و سرور نیست بعد از کرش سرور نیاز داره که فقط ری استارت بشه stateful اطلاعات رو نگهداری می کنیم به همین خاطر به حافظه نیاز دارند و به کلاینت گزارش از وضعیت اتصال می دهند، طراحی سمت سرور برای درست عمل کردن پیچیده تر و سنگین تر میشود بعد از کرش نیاز داره استیت های داخلی و همه اطلاعات رو سیو کنه.
ب) stateless، برای حل این مشکل از کوکی های سمت کاربر استفاده میکند و استیت ها رو اونجا نگه داری میکند و وقتی نیاز داشت به اطلاعات استیت هاش بازسازی می کند اطلاعات رو از کلاینت و سرور اون ها رو دریافت می کند.
یا از json web token ها استفاده می کنند.

سوال 3-

الف) بدلیل اینکه مدل کلاینت-سروری هستش مثلا سرور نمیتونه به سرور دیگه تقاضا کنه و جواب بگیره به دلیل پویایی اطلاعات حتما به api برای گرفتن از پایگاه داده دارد و نیاز به ادغام(شخص ثالث)، آوردن اطلاعات خارجی از api های

دیگر داریم مثلا لاگین کردن با اکانت گوگل مون یا فیسبوک تو یه سایتی مثل پینترست. در برخی کاربردها نیاز به مدل پابلیشر یا سرور مرکزی داریم (پوش سرور). Webhook ها با api فرق دارند، وب سرویسی که یه ایونت اتفاق میافته داده ها سبک رد و بدل می شوند تا اپلیکیشن در جریان اقدامات اعمال شده کاربر باشد.

ب) استفاده کردن از webhook ها برای در نظر گرفتن سناریوهای مختلف مکانیسمی مثلا برای پرداخت دوباره اگر سیشن تایم اوت شد. استفاده کردن از api ها برای داده های پویا، استفاده از سرور مرکزی (server push) برای پیاده سازی subscribe/publisher، استفاده از پروکسی ها.

سوال 4-

ss://asghar:1234!!@ss.myproxy.com:1234\#shadowSocks1
Ss→scheme
(Asghar→user 1234!! →password)
Ss→subdomain
Myproxy→second-level domain
Com→top level domain
#shadowsocks1 → به یه بخشی از صفحه

سوال 5-

1. 503 service unavailable / 403 forbidden
2. 401 unauthorized
3. 301 moved permanently / 404 not found
4. 413 payload too large
5. 200 ok
6. 403 forbidden

سوال 6-

1. reverse proxy
2. http tunneling
3. forward proxy