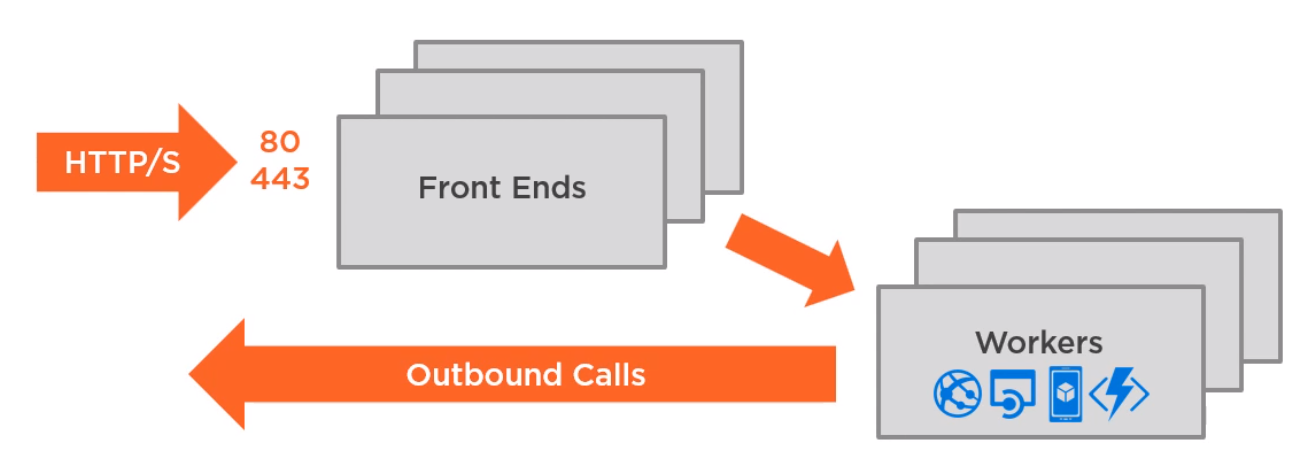
# App Services Plans

* Container for App Services
* Like a VM server farm
  + Region
  + Max number of VMs
  + Size of VM (CPI/RAM/Storage)
  + Pricing tier (Determines what is available and what isn’t) 🡪 note, cannot be changed after creating App Service Plan

1. Multi-tenant Service
   1. Workload isolation
   2. BUT: network infrastructure is shared with other Azure customers
2. ASE (app service environment)
   1. Deployed to my VNET, workload + network isolation
3. Azure Stack deployment
   1. On-premises

Every App Service:

* Has a C:\ drive (IIS, web app frameworks etc, temp files) and a D:\ drive (code is stored, logs, etc) 🡪 D:\ is in Azure Storage though, not local, this is where we FTP into
* Runs as a low-privilege worker process (called Application Pool Identity)
  + Even if two apps are running on the same VM, they can’t access each others memory, etc
* Code can call COM components
* Run PowerShell scripts, Open Command shell
* Access IIS, Application logs (but not Windows event logs)
  + Calls to Windows Event Log are faked by App Service Plan (XML file to the system drive)
  + No OS data, but good for debugging
* Only listen on port 80, 443
* Some outbound ports are blocked (445)
* Inbound-outbound IPs differ (front ends do load balance):
  + 
* Scaling happens: my app service plan grabs a VM from the „Regional Pool of VMs” and starts using it to scale

**ACU:** Azure Compute Units: expresses performance when comparing plans

No, you cannot upload precombined EXE applications as app services

**Let's say your applications are running on a Standard S1 app service plan, and you're starting to notice performance issues with your application.  You have a number of options for improving performance through scaling. Which of the following would NOT improve your application performance?**

Upgrade to Premium P1, because: The performance of the S1 plan and the P1 plan is basically the same.

**You create a brand new Windows Web App, and immediately deploy a Python application into it. The deployment works, but visiting the URL doesn't. The application works on your local machine but not in the brand new web app. What is required to get a Python app to run inside a Windows Web App?**

Python is disabled by default on the Web App, it has to be manually enabled in the portal