Azure Web Apps

Joshua Joshua

Joshua.Joshua@studentpartner.com

Azure App Service Family

Web Apps
Web apps that scale with your business

Mobile Apps
Build mobile apps for any device

Logic Apps
Automate business processes across SaaS and on-premises

API Apps
Build and consume APIs in the cloud

Azure Web Apps

- Support a variety of languages and platforms
  - .NET, Java, Node.js, PHP, Python, and more
- Support scaling (manual or auto) and load balancing
- Support slots for staged deployments and A/B testing
- Support continuous integration

Familiar and Fast

Leverage existing skills, plus languages, frameworks, and tools you're familiar with

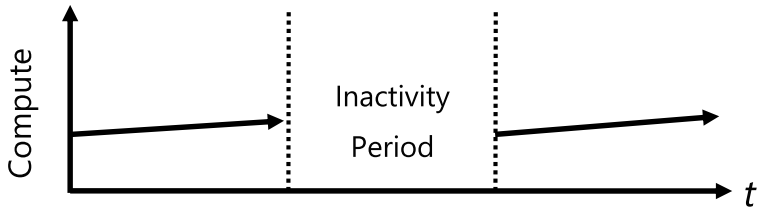
Enterprise Grade

ISO-, SOC2-, and PCO-compliant with enterprise-level SLAs

Global Scale

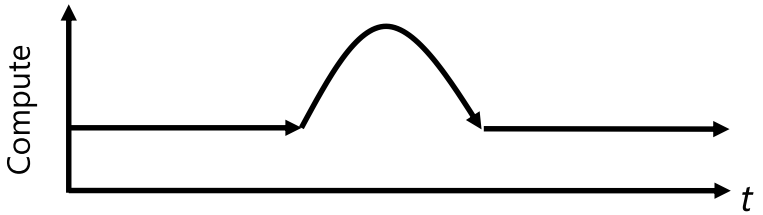
Scale up and down as needed, manually or automatically

Scaling - Cloud Computing Patterns



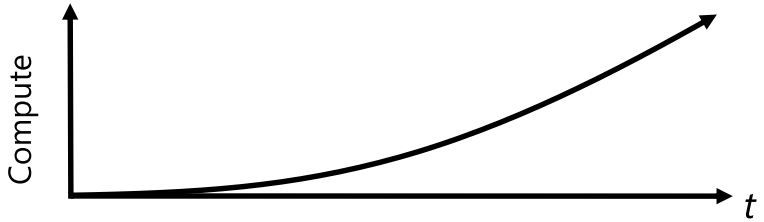
On and Off

On & off workloads (e.g. batch job)
Over provisioned capacity is wasted
Time to market can be cumbersome



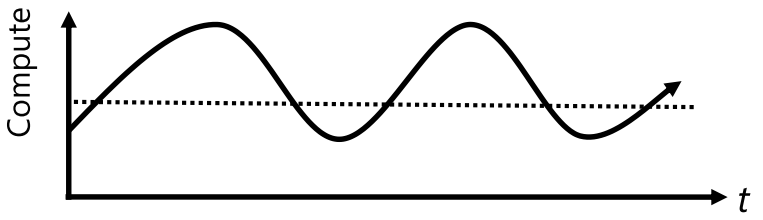
Unpredictable Bursts

Unexpected/unplanned peak in demand
Sudden spike impacts performance
Can't over provision for extreme cases



Growing Fast

Successful services needs to grow/scale
Keeping up w/ growth is big IT challenge
Cannot provision hardware fast enough



Predictable Bursts

Services with micro seasonality trends
Peaks due to periodic increased demand
IT complexity and wasted capacity

Scaling Up vs. Scaling Out

Scale Up

Vary the VM size

1 Core w/ 1.75 GB RAM
2 Cores w/ 3.5 GB RAM
4 Cores w/ 7 GB RAM

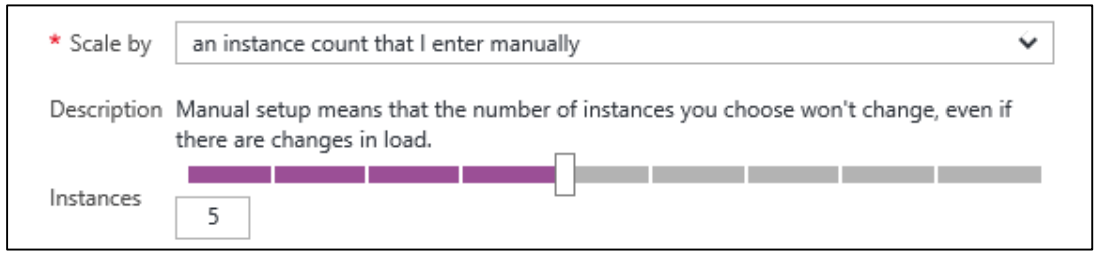
Scale Out

Vary the VM count

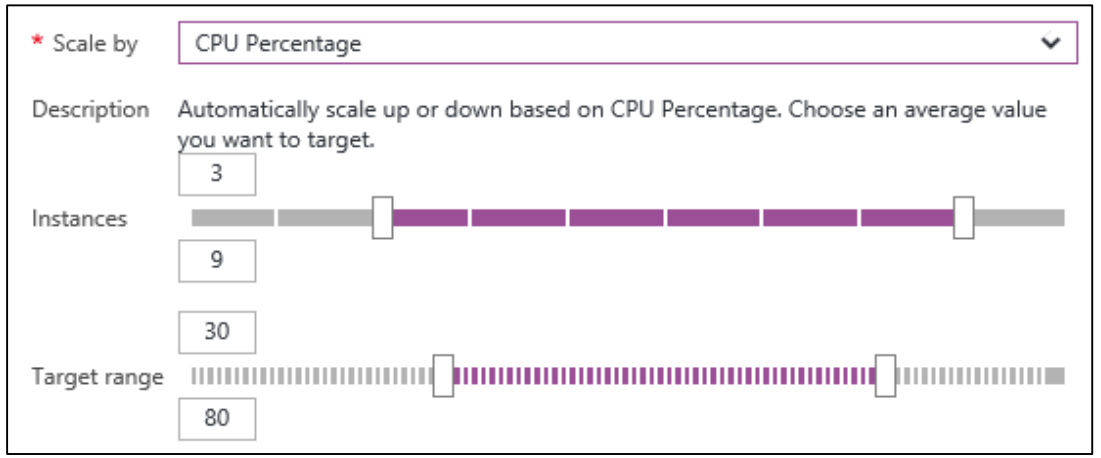
Max 3 instances*
Max 10 instances
*Max 20/50** instances*

Manual Scaling vs. Auto-Scaling

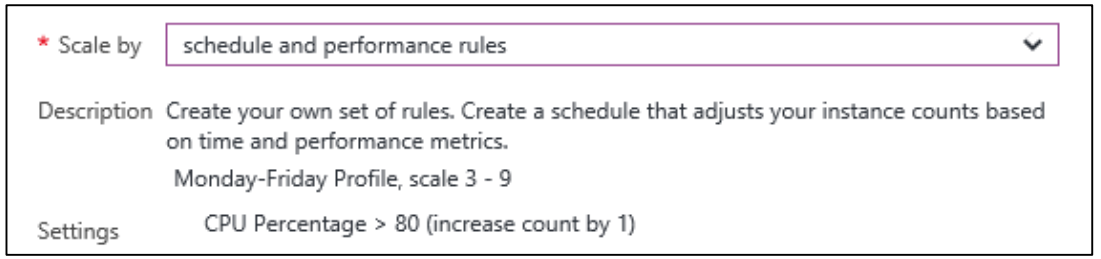
Manual – Scale via portal or scripts



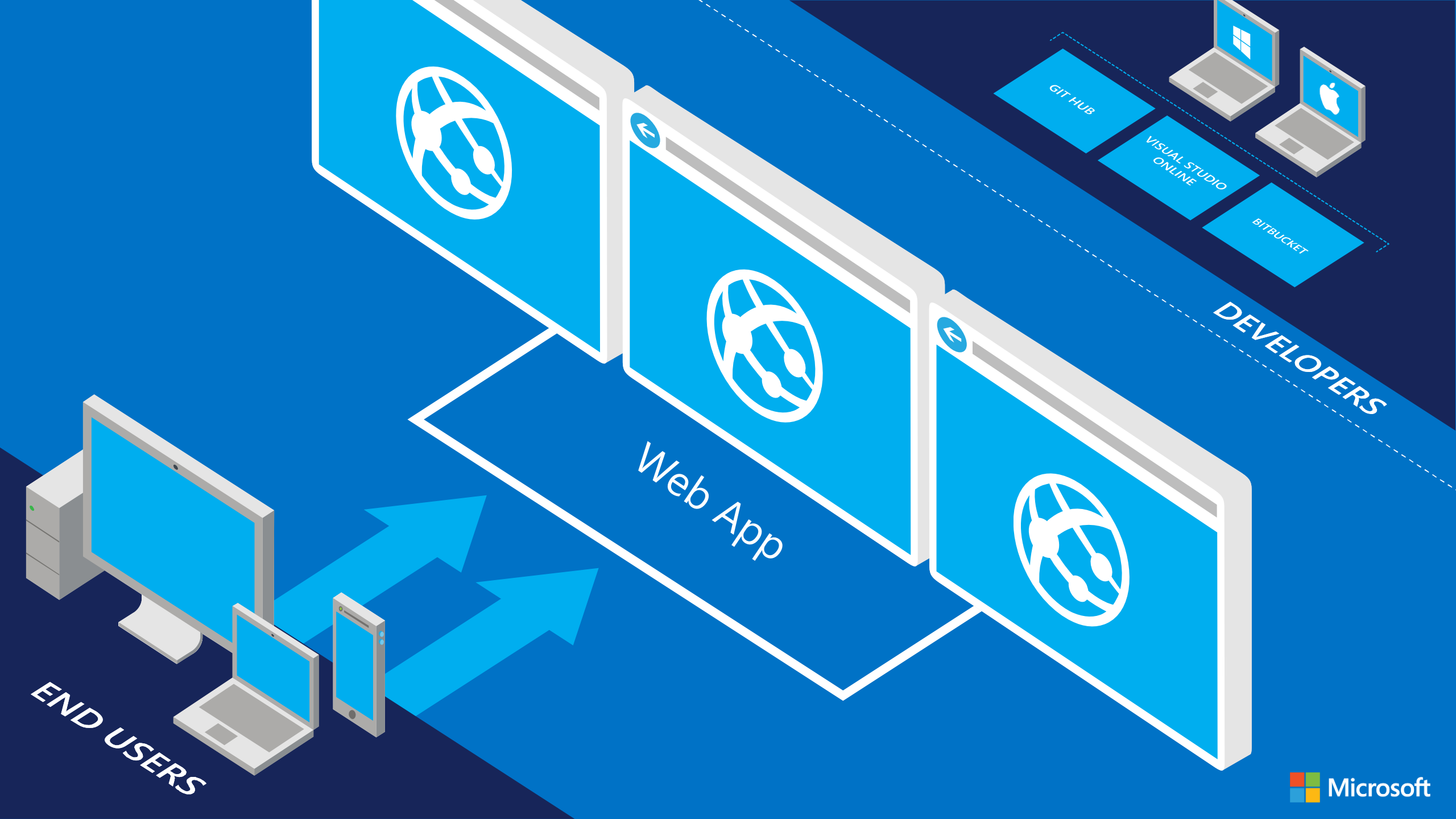
Auto – CPU Percentage

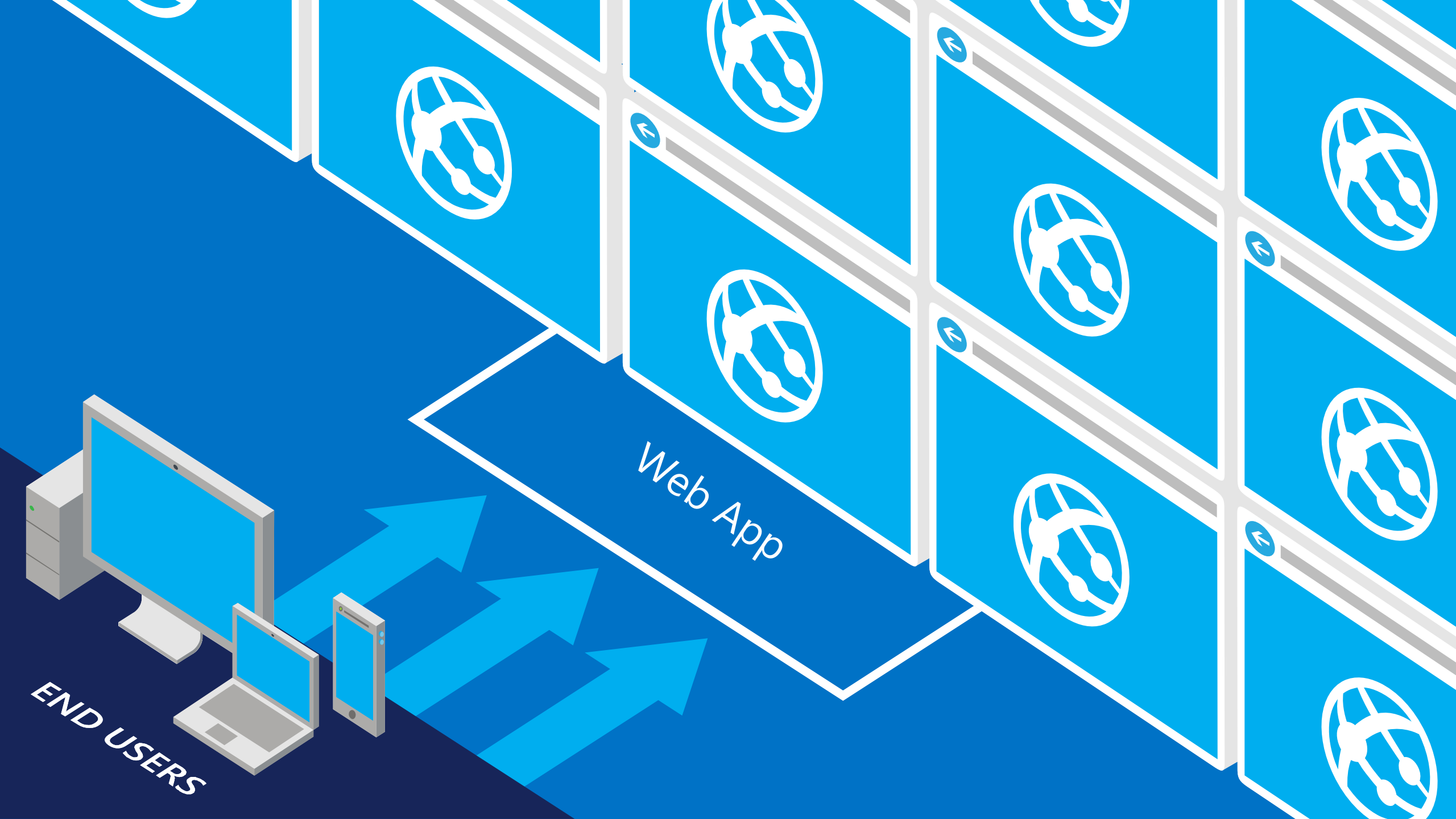


Auto – Schedule & Performance Rules



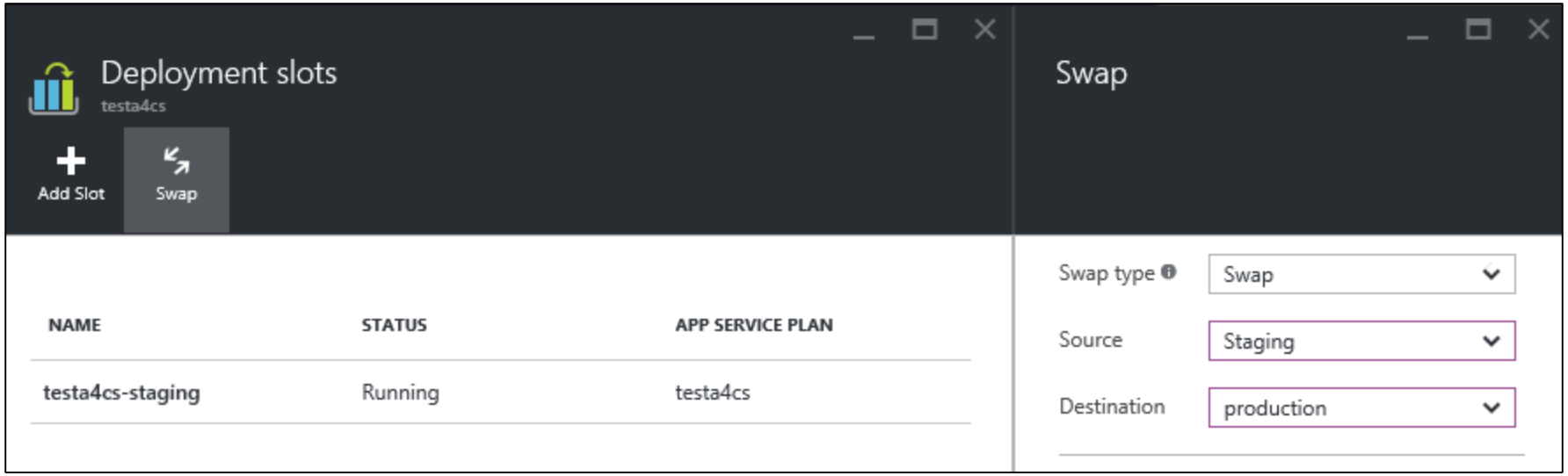

GIT HUB
VISUAL STUDIO ONLINE
BITBUCKET
DEVELOPERS
Web App
END USERS
Microsoft


GIT HUB
VISUAL STUDIO ONLINE
BITBUCKET
DEVELOPERS
Web App
END USERS
Microsoft


Web App
END USERS

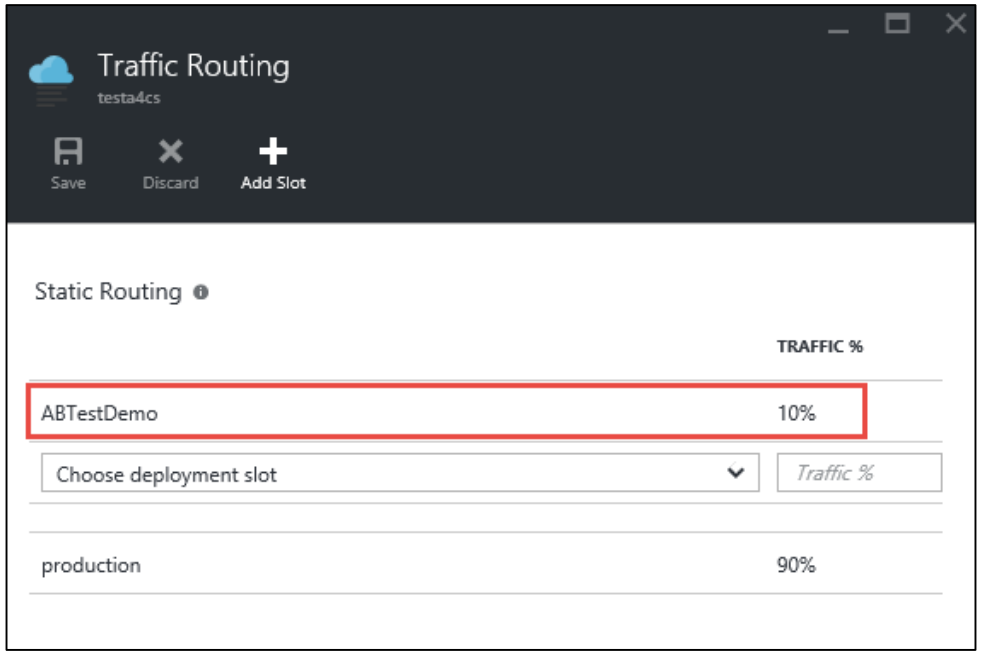
Deployment Slots

- Use a Deploy-Confirm-Promote workflow
  - Promote via “swap” through Azure portal
- http://sitename-slotname.azurewebsites.net



A/B Testing

- Test changes by routing requests to different deployment slots
- Use Traffic Routing to direct % of traffic to alternate slots



Continuous Integration

- Web apps can be deployed manually via FTP or WebDeploy
- Automate deployment using 3rd party source-control providers
- Can also use a local Git repository from Azure Portal


Git


Visual Studio Team Services

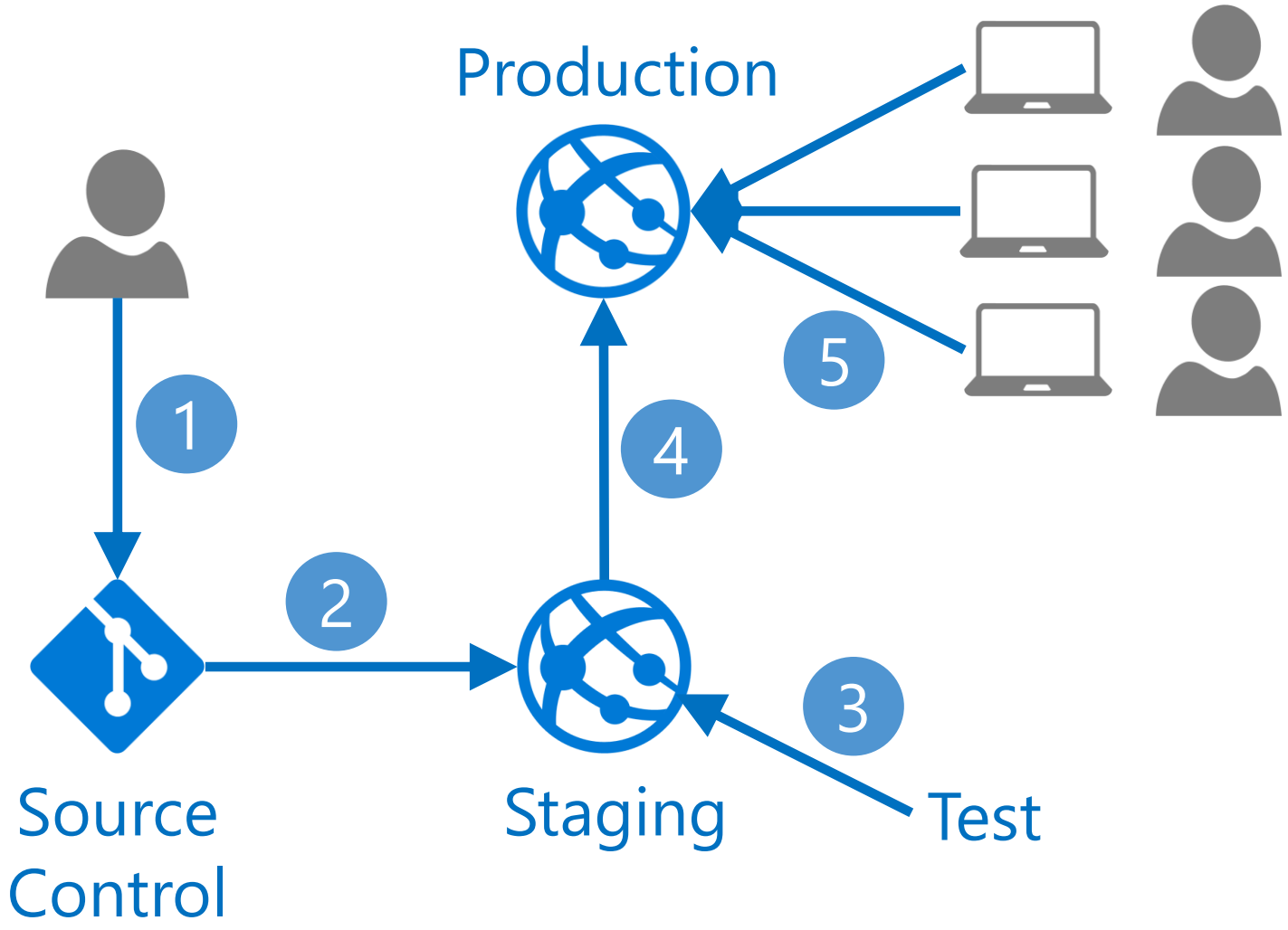

CodePlex


GitHub


BitBucket


DropBox

Continuous Integration + Deployment Slots



1. Developer commits code
2. Automated process builds/compiles and deploys to staging slot
3. Automated and other tests validate content in staging slot
4. Staging content promoted to production
5. Users see updated site

App Service Plans

- Billing and provisioning for App Service resources

	Free	Shared	Basic	Standard	Premium
# of Apps	10	100	Unlimited	Unlimited	Unlimited
Shared Disk Space	1 GB	1 GB	10 GB	50 GB	500 GB
Maximum Instances	1	1	3	10	50
Autoscale	No	No	No	Yes	Yes
Staging Environments				5	20
Custom Domains	No	Yes	Yes	Yes	Yes
SLA			99.95%		

Hands-On Lab

Azure Web Apps HOL.html