



Functionality	ParkMyCloud	In- House Scripting Tools	
		Small Environments	Large Environments
Multi-User/ Multi-Team	<ul style="list-style-type: none"> • Role-based access controls (RBAC) and access-based enumeration (ABE) for enhanced security • Single Sign-On - Okta, Ping, ADFS, Azure AD, etc. • Unlimited teams/Unlimited users • Cloud cost optimization laser development focus (compute, database, etc.) • Enterprise-wide visibility for automated on/off times • Options for centralizing or decentralizing control to departments, teams & individuals • Single view of all resources across locations, account and cloud service providers (CSPs) 	<ul style="list-style-type: none"> • Demands skilled DevOps personnel with knowledge of scripting & automation • Scarce DevOps resources distract from core business activities • Opportunity Cost • Significant risk if knowledge of infrastructure and scripting is managed by single individual (knowledge transfer) 	<ul style="list-style-type: none"> • Difficult to ensure consistency and standardization of automation approach across entire organization unless highly centralized • Complex and resource-intensive to provide DevOps support for all AWS environments across multiple teams/business units • DevOps resources distracted from core business activities - PMC offers API for integration into DevOps process • Opportunity Cost • Ability to devolve management of AWS instances to non-technical teams for scheduling on/off (PMC requires NO scripting) • Difficult to support existing team structures and ensure appropriate controls without building out complete custom solution
Multiple Credentials & Multiple Cloud Service Providers	<ul style="list-style-type: none"> • Unlimited number of credentials / accounts • Connect natively to limited access role in each cloud • Secure credential management (AES-256 encryption) • Multiple public CSPs - ability to manage AWS, Azure and Google for single platform 	<ul style="list-style-type: none"> • Security - Must develop means to securely handle and manage credentials and other sensitive account information • Multiple CSPs - Must develop approach and interface across multiple CSPs • Must keep up-to-date on changes to public cloud APIs and Services 	<ul style="list-style-type: none"> • Security - Must develop means to securely handle and manage credentials and other sensitive account information • Must keep up-to-date on change/updates to public cloud which is constantly evolving and adding and changing services. • Must develop approach to assign access to different credentials by different teams with PMC RBAC • Multiple CSPs - Must develop approach and interface across multiple CSPs



→ www.parkmycloud.com

Functionality	ParkMyCloud	In-House Developed Scripting Tools
Platform Coverage	<ul style="list-style-type: none"> • Instances/Databases/Scale Groups/Metrics • Logical groups - create, manage, and park by resource group • Single pane of glass view of ALL AWS, Azure, and Google Regions and Availability Zones 	<ul style="list-style-type: none"> • Scale groups - Must develop means to create a single view and the ability to manage and start/stop ASG's • Logical groups - Must develop means to create, manage and start/stop logical groups
"Always Off" Scheduling	<ul style="list-style-type: none"> • Temporarily suspend parking schedules during off-hours to enable ad hoc instance control • Use the UI or API for scheduling, parking, or snoozing 	<ul style="list-style-type: none"> • On-demand access - must develop a process to enable on-demand access to stopped instances in off hours • Schedule re-application - must be able to re-apply schedule when off hour work is done • Multiple accounts and CSPs - Must do this across multiple accounts and CSPs
Cost Visibility (Finance)	<ul style="list-style-type: none"> • Forecasts & displays future savings based upon selected schedules • Real-time actual month-to-date savings displayed • Generates & distributes ad hoc detailed cost and savings reports • Identifies underutilized resources 	<ul style="list-style-type: none"> • Visibility - need to develop custom application to determine cost savings based upon application of automation or removal of schedules (to date we have not encountered anyone who has developed such an application) • Reporting - would need ability for ad hoc reports over arbitrary date ranges
Automated Action	<ul style="list-style-type: none"> • Enterprise-wide policies based on tags to auto enforce actions (automate parking schedule assignment, Never Park for production instances, & assignment of instances to teams) • Utilization-based recommendations and automated actions to aid the discovery of cost optimization opportunities ---> SmartParking 	<ul style="list-style-type: none"> • Hard to enforce consistent and standardized policies within organization within decentralized structures where different automation tools are being used • Multiple CSPs - this would need to be done across all CSP accounts and across CSPs • Difficult to build something like Never Park or Snooze Only • Utilization data would need to be collected, aggregated, stored, and acted upon
Cloud Operations	<ul style="list-style-type: none"> • SaaS, 15 minute setup, no scripting or coding • 6 week payback/ROI • Admin can grant rights to Team Leads with RBAC • Multi-cloud/Multi-account/Multi-user 	<ul style="list-style-type: none"> • Multi-account/user - Scripting typically doesn't support multi-cloud / multi-user / multi-account access • Does your scripting: <ul style="list-style-type: none"> ◊ support more than instance / vm start and stop - databases? scale groups? usage metrics? ◊ integrate with your CI/CD pipeline? ◊ capture savings data for report distribution? • Schedule override - Typically does not let users 'snooze' schedules when they need to access them while scheduled • Sequential start/stop - Typically does not let users group resources and start / stop sequentially

