MILARNE CONSULTING

DRIVING IT INFRASTRUCTURE RELIABILITY

IT Infrastructure Transformation

CTO Guide: From Zero to Hero

What is Infrastructure Transformation

Whilst *the millennials* in IT departments are building digital apps, many companies still have legacy infrastructure and data centres that manage core systems that *Gen Xers* need to maintain

These core systems often run on old and unsupported infrastructure

Let's not forget these systems and the need to maintain them

An IT Infrastructure Transformation program will consider:

Where we are at now	What the ideal future state for us will look like and how can we avoid repeating the same mistakes	What options do we have to uplift or transform our infrastructure

MILARNE CONSULTING
DRIVING IT INFRASTRUCTURE RELIABILITY

Infrastructure Transformation

A current state assessment is often performed *bottom up* but a desired state assessment is often *top down*

- Where are we now (bottom up)
  - Still running:
    - High cost legacy data centres
    - Old hardware, out of maintenance
    - Unsupported operating systems
    - Unsupported Databases versions
    - Unsupported middleware
  - Inadequate Disaster Recovery
    - Many apps have no DR
    - DR tests are simulated, not real
    - Can fail over but can't fail back
    - DR Plans that don't work
  - IT spend is too high
    - Extended maintenance
    - Staffing to manage old systems
    - Reliance on 3rd parties who may have the same issues
  - Vulnerability and Risk
    - Inability to maintain unsupported infrastructure
    - Exposure to adverse events, many single points of failure
    - Inability to maintain SLAs and the flow on business impact

- Where do we want to be (top down)
  - Support the business at all times, ensure the quality of infrastructure service meets business requirements
  - Get business buy in on infrastructure support
    - The cost of new projects could/should include maintaining the infrastructure for it's lifetime
    - Don't let infrastructure become out of support
    - Does the business know their DR probably doesn't work?
    - Balance new project spend with reliability & resilience spend, focussing on critical systems
    - Sometimes it is ok to accept the risk
  - Robust IT infrastructure
    - IT Operations is all about 'keeping the lights on'
      - Reliability (the light always comes on when you flick the switch)
      - Resilience (the ability to mitigate, survive and/or recover from high-impact adverse events)
    - Continuous improvement and maintenance
  - Modernise and maintain
    - Remove costly data centre real estate
    - Keep infrastructure current and maintained
    - Focus on Disaster Avoidance for critical systems, not Disaster Recovery. Engineer resilient systems
      - The cost to build resilience is not much higher than the cost to build recoverability

How to we get from where we are now to where we want to be

Remove physical hardware

Transform/Rationalise and decom legacy systems

Consider migrating to the Cloud

Carefully assess which cloud provider(s) and what apps

Remove Physical Hardware

- Perform an audit of your data centre(s)
- Much of it will be old and running unsupported OS and DB versions
  - Why can't it be virtualised?
    - The reason why not back in the day may not apply now
      - For example, SQL and Oracle now run well on virtualised infrastructure which is way more powerful than it was when these legacy systems were deployed
    - All it will often require is a *shallow dive* with the apps teams to identify opportunities
  - Look hard for decom opportunities
      - When the business say no, ask 'why not' and 'what has to be done to decom'
        - Maybe it won't be so hard

Transform/Rationalise and decom legacy systems

- **3 key options on-prem:**

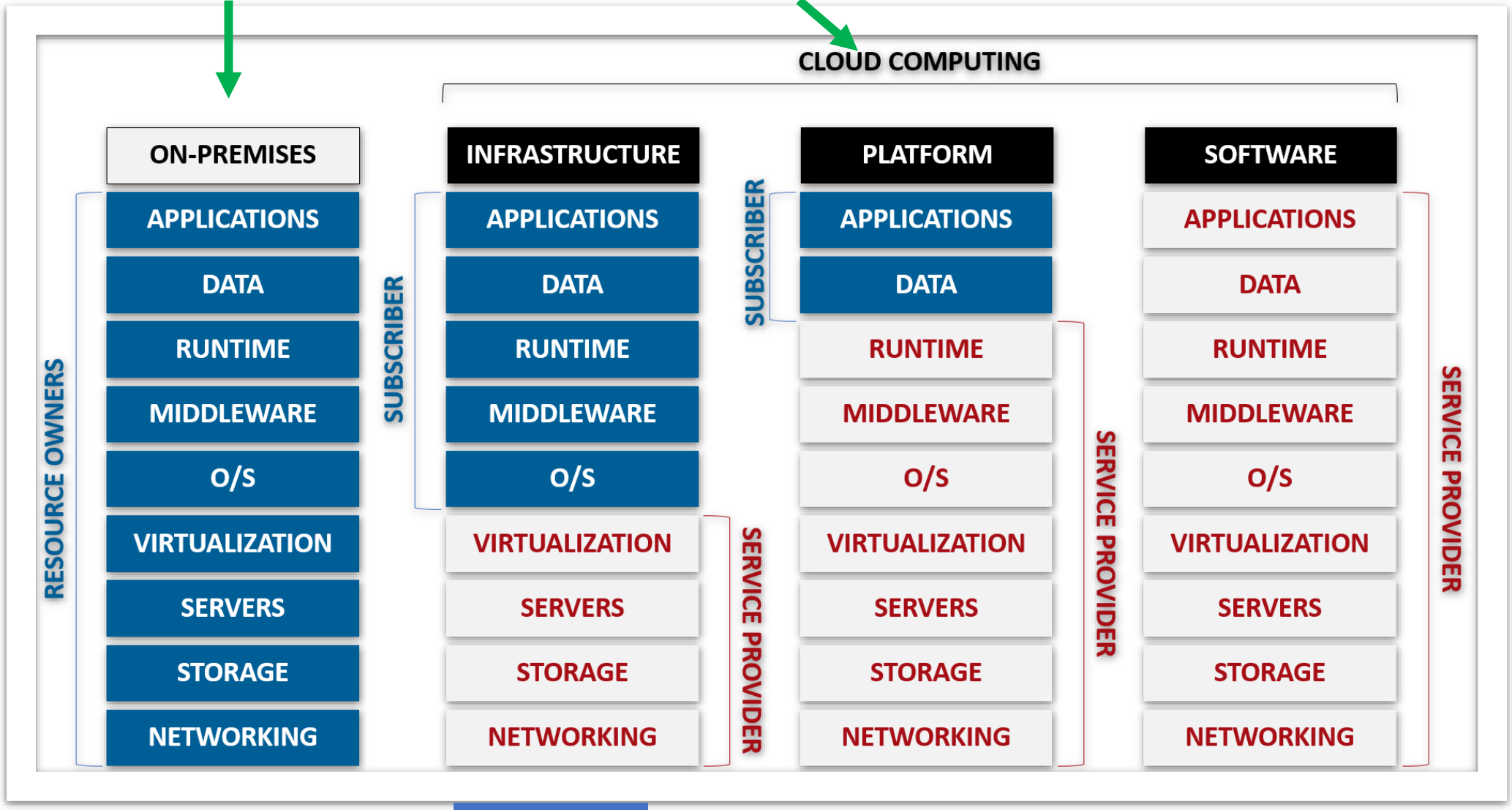
No.	Option	Strengths	Weaknesses
1	Modernise app & underlying infrastructure	• Fit for the future • Support, compliance & reliability	• Cost: Will often require apps code and middleware changes as well
2	Infrastructure uplift 1) P2V 2) Technology uplift (OS & DB)	1. Removes old hardware 2. Provides support, compliance & more reliability	1. Does not uplift OS and DB etc 2. More expensive. DB upgrade may need apps changes
3	Decom/Rationalise. 1) Remove duplicates (e.g. batch schedulers) 2) Decom dead systems	1. Less cost & complexity 2. Less cost	1. Cost to execute 2. Cost to execute

- Don’t forget DR and that Non-Prod and Prod environments must be as close in configuration as possible, particularly Pre-Production Test Environment
- **Off-Prem-Migrate to the Cloud**

Migrate to the Cloud

- Cloud 101
  - Move from a Capex model of buying, deploying and managing physical infrastructure to an opex-based *shared responsibility model* using infrastructure from a cloud vendor that:
    1. Is Well-Architected, reliable and resilient
    2. Provides capacity and performance on demand
    3. Is very quick to deploy
    4. Is scalable and elastic to match the ebbs and troughs of business
    5. Engineered to the very best industry standards that meets industry and government compliance requirements
- How many of us can say our on-prem systems do all this ?
- Lets look at 2 key elements of Cloud Computing
  - The Shared Responsibility Model and
  - The Well Architected Framework

The Shared Responsibility Model: How Infrastructure and Applications are managed On Prem and in the Cloud



In the Shared Responsibility Model, On Premise indicates that it is the responsibility of the resource owner or its delegate to manage. Cloud Solutions include *Infrastructure/Platform* and *Software* as a Service with management responsibility of resources shifted to the provider.

Well Architected Framework

Pillar Name	Description
Security	The ability to protect information, systems, and assets while delivering business value through risk assessments and mitigation strategies.
Reliability	The ability of a system to recover from infrastructure or service failures, dynamically acquire computing resources to meet demand, and mitigate disruptions such as misconfigurations or transient network issues.
Performance Efficiency	The ability to use computing resources efficiently to meet system requirements, and to maintain that efficiency as demand changes and technologies evolve.
Cost Optimization	The ability to avoid or eliminate unneeded cost or suboptimal resources.
Operational Excellence	The ability to run and monitor systems to deliver business value and to continually improve supporting processes and procedures.

- Both AWS and Azure have a Well Architected Framework that aligns closely to this

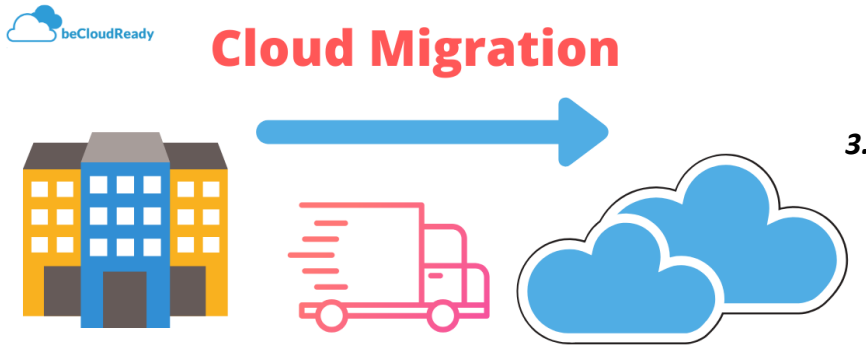
Migrate to the Cloud

Remember our bottom up approach from earlier:

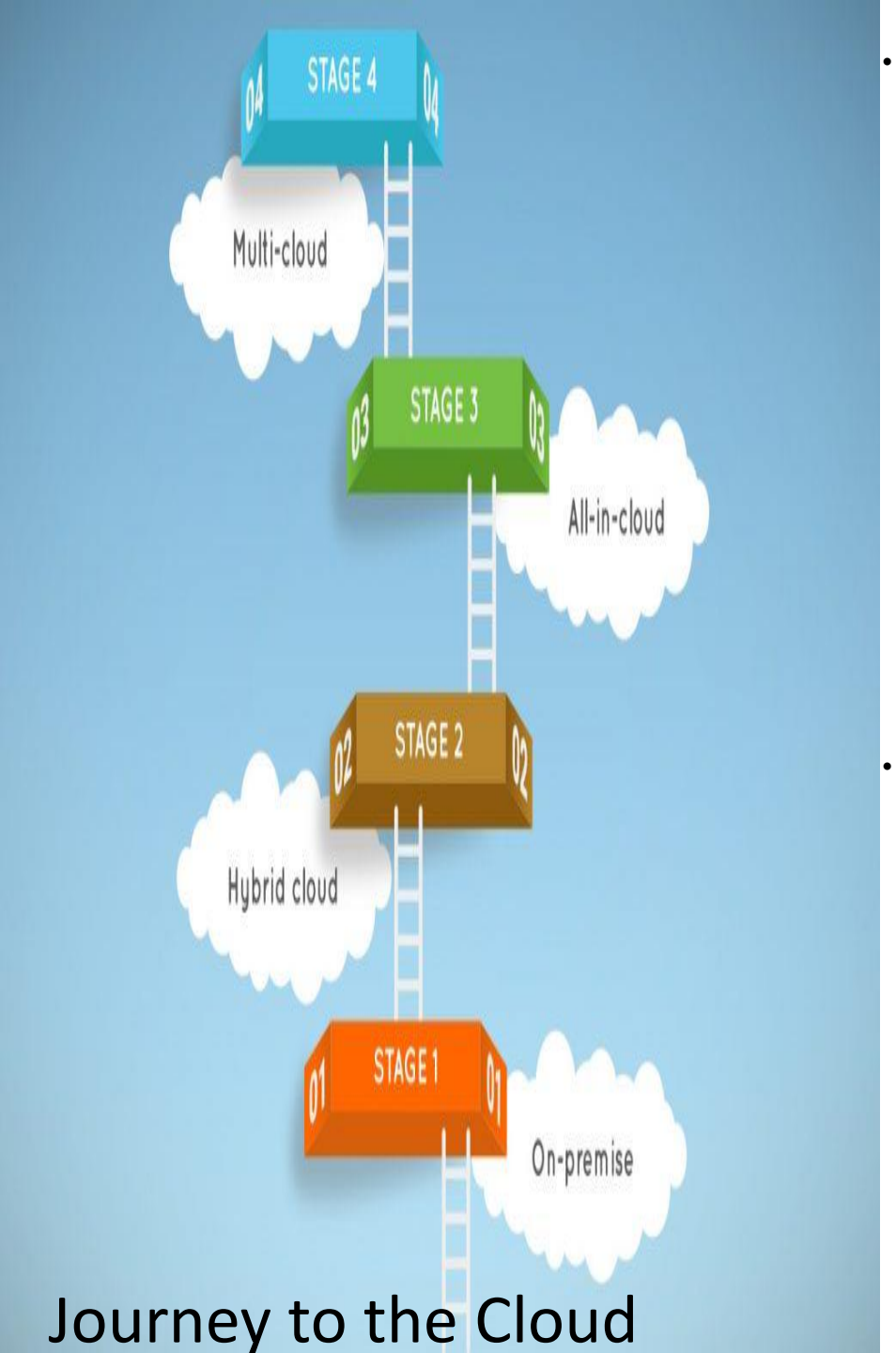
- Where are we now (bottom up)
  - Still running:
    - High cost legacy data centres
    - Old hardware, out of maintenance
    - Unsupported operating systems
    - Unsupported Databases versions
    - Unsupported middleware
  - Inadequate Disaster Recovery
    - Many apps have no DR
    - DR tests are simulated, not real
    - Can fail over but can't fail back
    - DR Plans that don't work
  - IT spend is too high
    - Extended maintenance
    - Staffing to manage old systems
    - Reliance on 3rd parties who may have the same issues
  - Vulnerability and Risk
    - Inability to maintain unsupported infrastructure
    - Exposure to adverse events, many single points of failure
    - Inability to maintain SLAs and the flow on business impact

- Does a Cloud solution meet these objectives ?
  - A properly architected cloud solution for legacy apps will:
    - Help remove legacy datacentres
    - Remove old hardware and out-of-maintenance issues
    - Remove unsupported operating systems or allow them to be supported for longer
    - Remove Unsupported databases
    - It may not address middleware
  - If a Well-Architected Framework is deployed, a resilient environment that is built to expect failure will avoid almost all unplanned outages and provide the ability for live DR/failover/failback and to perform maintenance without impacting applications availability
  - Disaster Recovery will be replaced by Disaster Avoidance
  - Do not assume a net cost saving but assume a move from Capex to Opex model
  - Staff may move to DevOps roles
  - Significantly less exposure due to the best monitoring, alerting and preventative tools in the business

Top 5 reasons to migrate legacy apps to the cloud



1. ***Always on: Exploit the Well Architected Framework***
   - Resilience out of the box
   - Regions, Zones, Edge locations, global services
   - Plan for failure approach
   - The services where the cloud provider is responsible will always be current
2. ***Services are best-of-breed/utilise industry best practices***
   - Security and Threat prevention
   - Network design is very robust and always assumes *no access* as the default
   - Identity Management
   - Database scaling and performance
   - Very compliant with industry & government regulations
3. ***Speed to Value***
   - Build test environments in hours
   - Supports Fail-Fast, Agile delivery
   - Once the framework is in place, Prod infrastructure deployments can be in days or weeks rather than months
   - Removes the capex delay, e.g. PO approvals, hardware delivery lead time, installation and hardening etc
4. ***Performance: Auto Scaling, Elasticity, Front-end caching***
   - Scale up and down as required
   - Spread load in the event of capacity or availability constraints
   - Edge locations and caching offerings enhance end-user experience
   - Serverless capabilities
5. ***Cost Saving Opportunities***
   - Non Prod (spin up and down)
   - Big Data (DWH, ETLs, Media files etc) offerings are PAYG rather than having to spend millions on infrastructure
   - Platform and Software as a Service options remove all of the infrastructure management overhead
   - Possible license savings (each use case is different)



Journey to the Cloud

- ***Which Cloud Provider***
  - Talk to each vendor, letting them know the competitive landscape. Build an assessment/decision matrix that includes
    - Hard factors such as: BAU cost, Migration cost, Software licensing, Skills uplift
    - Soft factors such as: ease of use, cultural and business priority alignment, support when you need it (e.g. how hard do they help you make the pilot successful)
  - Apply weightings to criteria and score each vendor. Be objective, avoid reverse engineering the outcome
  - AWS are good at pitching to Execs. Execs are not in the trenches. Ask decision makers to be objective
- ***Check the numbers***
  - Run the cloud-provider business case/cost comparison macros but don't assume that the cloud will be as cheap as the numbers tell you
  - Azure Enterprise License or AWS Reserved Instances are much cheaper
  - Avoid: Idle instances, under-utilised reserved capacity, unattached volumes
  - Be careful which storage tier you chose as the faster the IO or throughput, the more expensive it is.
    - You may not need super-fast, particularly for legacy apps
  - Tag resources so you know who is using them
- ***Get the right team***
  - Need a combination of legacy infrastructure SMEs and cloud SMEs
    - Architects, PMs, Engineers
  - Be careful which 'consultants' you engage. Do they understand legacy app topology ? Are they outcome focussed ? Will they be your advocate? Will they add value or just go through the motions?
- ***Run a pilot with some legacy apps***
  - Linux and Wintel based, old and new OS use cases
  - Check sociability and performance (against an on-prem baseline) back to on-prem/legacy apps and databases
  - Run pilot via VPN connection as it will take time and money to establish AWS Direct Connect or Azure Express Route
- ***In parallel:***
  - Prepare your version of the Well Architected Framework
    - It must be setup at the start as it will be much harder to retrofit capabilities later
  - Decide how to migrate servers (cloud provider tools or 3rd party tools such as Zerto)
    - Test this in the pilot
  - Consider exit strategy/vendor lock in/vendor relationships/strategic partners cloud vendors

Which cloud provider (of the top 2) ?

- Linux (open source) oriented, engineer-centric, build the blocks one by one
- Configuration flexibility and customisation
- Region/Availability Zone
  - Region-geographic location with multiple AZs
  - AZ - 1 or more data centres
- 1 Availability Zone (not 2) in Australia with closest AZ in Singapore for multi-AZ architecture
- Very strong DBaaS offerings
- Several wizards such as Elastic Beanstalk
- Many Big Data offerings
- Several storage tiers
- Comprehensive 3rd party integration (e.g. AWS Marketplace)

- Microsoft oriented, less engineering focus, less build work but less config flexibility
- Geography/Region/Availability Zone.
  - Geo area containing at least 1 region
  - Region-Set of datacentres within latency defined perimeter. regional Data Centre pairs
  - Zone. Unique location within a region. Could be in same city
- License saving opportunities on Microsoft products including on-prem
- Integration with other Microsoft products that you are probably using (e.g. SQL, Outlook, SharePoint)
- Hybrid-cloud centric
- No data charges intra-region
- Ability to monitor, manage and integrate to on-prem as well including natively with Active Directory
- Strong Compliance and Regulatory alignment

My recommendation for legacy apps as long as pricing, performance and pilot results are encouraging is Azure because:

1. Most legacy infrastructure farms are > 66% Windows based
2. Microsoft want market share and will offer better pricing and support including for on-prem services. They will probably also continue to support very old operating systems (e.g. Win2003)
3. Azure has Availability Zones in Melbourne and Sydney
4. They are hybrid-cloud-centric, less 'cloud or nothing'
5. AWS are more arrogant and less customer-first.

Summary

- A CTO responsible for legacy IT infrastructure can move from Zero to Hero by adopting a sound transformation strategy
- Migrating legacy applications to the cloud is very achievable but the approach and execution process will have a say in the success of the outcome and the cost savings that will be anticipated
- Good Luck !

Zero to Hero: To change an outcome, one's situation, or oneself from being particularly unsuccessful, negative, unfortunate, or unpopular to being especially successful, positive, fortunate, or popular.