What are Your Cloud Transformation Principles?

Jonathan Allen, Enterprise Strategist & Evangelist at Amazon Web Services

“Taking a small amount of time to define the principles for your cloud adoption will save you incalculable time.”

Tenet: (n) A principal or belief, especially one of the main principles of a religion or philosophy. These tenets are held by the group. To be a member in good standing, you are expected to hold to its particular teaching.

In my role as an AWS Enterprise Strategist I travel the world working with leaders from the largest organizations. A continual leadership challenge I hear is how to achieve consistency of action and commonality of judgement with folks across the business. It’s nice to think that everyone starts a transformation journey at the same logical time and place, and it’s easy to assume everyone understands “the why” and “the how” of that transformation. But even leadership teams that are seemingly aligned (and I have sat on many) often “think” they are agreeing on the scope of the same problem but never actually have the conversation to align properly and calmly on the why and the how in the clear contextual map of the world their business lives in.

One of the mechanisms used at Amazon to deal with potential misalignment and ambiguity head on is its effective use of principles (called tenets, internally). A team defines its own tenets, and these tenets are challengeable by any member of the team. If any members thinks they know a better one, then they have the responsibility to recommend a change. Taking a small amount of time to define the principles for your cloud adoption—and to consciously allow those principles to be challenged and amended as you know more—will save you incalculable time.

I have personally experienced and witnessed individuals who have spent days, weeks, and months (and yes, even years) building software and infrastructure that seemed important to them only to find it’s not used by anyone. One of the most common examples in our industry are “data lakes”. I have lost count of how many enterprises I have worked with that had a team working away in a silo, constructing a thing of beauty around an on-premises Hadoop and storage only to find no one places their data in it or uses it for answers. Such a waste. A simple agreed ‘principle’ constructed around data could have prevented this. Something as simple as “Each team owns their data and makes it securely accessible on a need by need basis by everyone via an API” could have solved massive wasted time and energy.

Organizations that take the time to agree upon and declare their principles have four things in common: 1) the declared principles are heavily used at the macro level to make decisions, which avoids valueless creation and needless escalation; 2) their organizations consistently achieve greater flow (more releases), as decisions are made faster; 3) the principles are used at all levels to drive consistency of judgement and action; 4) the principles are used as road signs for your cloud transformation.

Here at Amazon, the Amazon Leadership Principles (LPs) drive every single Amazonian to use their judgement consistently on a massive scale. I can have a conversation with any employee on any day about "Diving Deep" on a customer problem or not turning up for an internal meeting because LP1 (Customer Obsession) required my undivided attention. These principles are designed to be naturally contentious, to provoke a conversation. Every single day every Amazonian is reflecting on them and balancing them. Sometimes we have to "Dive Deep" before showing "Bias for Action". Sometimes we just need to make things happen. How you use the principles requires judgement, and we recruit strongly for that judgement in respect to these principles.

So, what makes a good cloud principle?

Well, my good colleague Joe Chung posted a great blog on this subject going into some awesome examples. But I want to take a moment to build on this great article. Over the last year alone, I have helped over 147 organizations from around the world with their cloud journey, and I've come across a number of different principles. While this list is by no means exhaustive, I wanted to provide a list of examples that anyone could copy, amend, and use as they think appropriate and apply to their organization. I've organized this sample list of principles around the five principal benefits of why enterprises move to the cloud: **security, cost, flexibility, compliance, availability, and people**.

Security

- **Source Code Security:** All code will be securely held in central code repository. Access will be monitored.
- **Policies Matter:** While teams have autonomy to choose their tooling, the tools and solutions must comply with security and availability objectives.
- **Credential Blast Radius Reduction:** We will appropriately reduce access to the minimum.
- **Assume the Enemy Knows Your Code:** Dance like no one is watching, encrypt like everyone is.
- **Radically Restrict and Monitor Human Access to Data:** Drive people to use tools to access data rather than by hand.
- **Immutability Rules:** The authoritative data source and logs will be immutable. A copy of data will be held separately from the team(s) that supports the data.
- **Trust but Verify:** We will intrinsically trust our leaders, engineers, and developers to make the right decisions to protect our data and systems, but we will have mechanisms in place to verify that trust.
    - There are a number of AWS tools and great partner solutions that can help with this, including Turbot, Stratus Cloudtamer, CloudCkr, Saviynt, CSRA Agility, Red Hat, Telos, Evident, Cisco Cloud Manager, CloudAbility, Fugue.io, and DivvyCloud.

"

How you use the principles requires judgement, and we recruit strongly for that judgement in respect to these principles."

"

We acknowledge that 'perfect' is the enemy of 'good enough.'"

Cost

- **Cloud First:** To remove as much undifferentiated heavy lifting as soon as possible, all new development will be cloud native.
- **Cloud Native:** Wherever possible, we leverage AWS features rather than build our own solutions. We build the thinnest possible control plane over AWS to leverage their efficiencies of scale. We acknowledge that "perfect" is the enemy of "good enough". While we are biased to using AWS features, when blocked we will innovate with our own temporary solutions.
- **Run Less Software:** If a component has become a commodity, we shouldn't be spending precious development time on maintaining it, instead we should be consuming it as a service.
    - In the history of enterprises this is controversial, but even containers are now run and operated as a service. If your engineers aren't building data centers any more, why are they building container platforms?
- **Focus on Customer Data and Logic:** We strive to build and support the company's data and logic structures, not systems that do not differentiate our product.
- **Predominant Public Cloud Partner:** We will select a cloud partner who will allow focus for our organization to get to an expert level rapidly with a chosen platform, avoiding the distractions that come with too many platforms across people, process, and technology paradigms.
- **Minimum Viable Product/Cloud:** We will investigate the minimum security, availability, and efficiency objectives to get the first production workload to the cloud. We will expand our research to other tools as customer features demand it.
- **Close Data Centers by Set Date (Burn the Ships):** We will have migrated or found the right homes for all our systems to enable the close of our data centers by a specified date.
    - Setting a deadline in the sand, while a little daunting, creates momentum, focus, and gives everyone an irrefutable target. In 2015 when Rob Alexander (Global CIO) stood on stage at AWS:Reinvent and told everyone in the world that Capital One was closing multiple data centers and migrating to AWS Cloud, it was an unambiguous example of a declared goal.
- **Save as you Earn:** The team and product manager are accountable for their cloud spend if a means to an end justifies the use of something that delivers material fiscal benefit to the organization they are allowed to use.
- **Frugality Matters:** Being prudent and owning our cloud spend is important. Teams should strive to continually lower costs. Money spent on wasted resources could have been better spent on customer features.

Flexibility

- **Two Pizza Teams:** We will organize ourselves into small teams no larger than 12. Wherever possible, the teams will be self-contained and have the ability to own their destiny and work schedule.
- **You Build It, You Run It:** As the new small teams create features, they will own the support of them 24/7/365. DevOps in its simplest form.
    - It is often at this point that I get asked about segregation of duty. Ultimately, you want all code deployments and operations to be highly automated, using code pipelines for everything. If you need to retain two-man control, then consider differing levels of roles within one team and/or two-man automated role approval for pipeline deployment.
- **The Team We Have is the Team You Need:** We are always working to re-skill, retool, and promote our workforce with the best knowledge so they can execute our cloud vision first before trying to hire externally.
- **Teams Choose:** The team, with their product manager, decides how to build and which tools to use, as long as they meet the company's security and efficiency objectives.
- **One Size Doesn't Fit All:** Our business is large and diverse. Use the right tool for the job. We do not assume one size (tool or product) fits all, but we do have strong opinions on how to solve common problems. We automate and codify out opinions into simple, integrated experiences. We remove and deliberately avoid undifferentiated engineering effort.
- **Get Out of the Way:** Allowing service teams to own their AWS adoption themselves, we decouple and decentralize development. We prefer to build guardrails, not gates. We automatically audit for compliance.
- **Publish Your API:** Publishing your API each time will ensure the product and data is accessible securely via an internal, and if appropriate, external restful-API.

"

People are most likely to follow those they understand"

Compliance & Availability

- **Everything Fails All the Time, Design for It:** We will design and test for failure to levels appropriate for the customer problem we are solving. We will use site reliability engineering principles as we go, which is second nature to us.
- **Fail in Production:** We will be bold and use chaos engineering to deliberately fail components in a controlled way.
- **Production Always Runs in Multiple Availability Zones:** Production services and its data are always run in more than one availability zone.

People

- **Everybody is a Security Engineer:** Everybody focuses on appropriate security every day.
- **Pair Programming Works:** For both training and development of production code and support, we will frequently practice the use of two developers working side by side on a single machine. The sum is greater than the parts.
- **Tooled Correctly for Continually Learning:** We will ensure developers have the tools they need for the job, and we will put in place mechanisms to encourage and reward continual technical self-development.
- **Certification Rules:** We will encourage, recognize, and reward those engineers and developers who attain AWS certified status.
- **Get to 10%:** Getting 10% of all engineers and developers certified is our goal.
- **Recruit for Alignment with our Tenets:** Ensure that the folks you want to hire have demonstrable experience aligned to them.
- **Recognize what Motivates Engineers and Developers:** Motivation comes from autonomy, mastery, and purpose—allowing people to run with their own ideas, master them, and have impact with them.

About the Author

Jonathan Allen is an Enterprise Strategist & Evangelist at Amazon Web Services, working with enterprise technology executives on strategies for how the Cloud can advance their business.