Guide to Tagging Cloud Deployments

April 2020

v1.0.0

Table of Contents

Acknowledgements

Authors
Stephen Kuenzli, k9 Security

Contributors and Reviewers
Steve Sutton, k9 Security
Will Button, willbutton.co
Elliot Murphy, KindlyOps
Jeff Nickoloff, Topple

Introduction

Resource tagging is the primary context communication method in the Cloud. This guide helps technology teams tag Cloud application and infrastructure resources with the context needed to manage, operate, and secure those resources effectively.

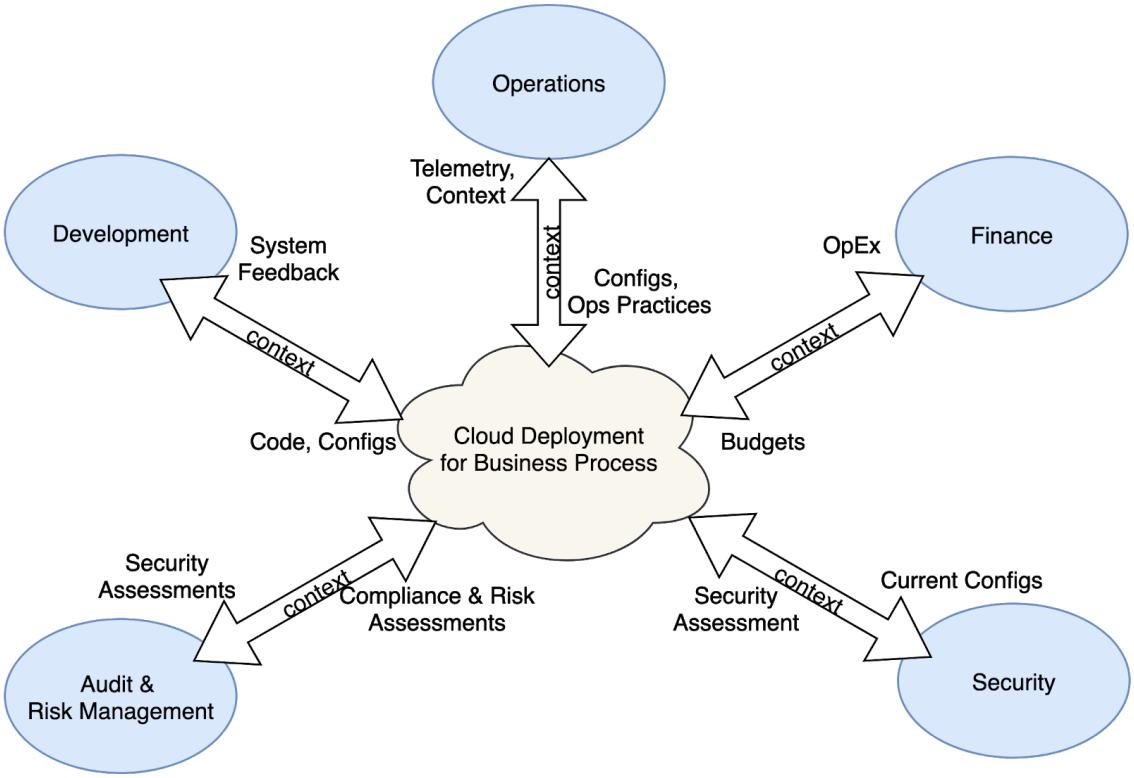

Figure 1: Cloud Deployment and Supporting Functions

Supporting several complex applications without explicitly modeled context is difficult for people and might be impossible for tools. People outside of development and operations may have great difficulty answering basic questions about deployed applications. This is often the case when analyzing cost, security, and risk.

The tags described in this guide that will help you answer questions like:

- Who owns this resource? What application does it belong to?
- Who should we call when the application is broken?
- Who should pay for this resource? Which applications are driving our costs?
- Do access controls secure this resource appropriately?
- How much risk does our Cloud deployment have? Where is that risk concentrated?
- Which security improvements reduce our risk the most?

Answering these questions requires ***context***. Once you have that context, you can make better decisions, quickly.

Example: Should an S3 bucket be publicly accessible? Depends on its ***intended*** use.

The correct answer is 'yes' when the bucket hosts a public-facing customer website and 'never' when storing customers' personal health information.

Both people and tools rely on:

- identifying resources individually, as groups, and the relationships between them, e.g. compute instance X is a member of cluster Y supporting application Z
- scoping resources to a particular management, fault, or security domain, e.g. an Environment
- describing important attributes of the resource's responsibilities, capabilities, and lifecycle, e.g. a Role within an Application or intended Confidentiality level

The organization can analyze deployments and *answer their own questions* when resources are identified, scoped, and described well. This is a much more efficient and scalable alternative to chasing down engineers on the application, platform, or security team to answer those questions. Further, the organization can standardize management of contextualized Cloud deployments.

Tagging schemes are the primary way Cloud deployments communicate context to the organization. Cloud providers and many tools use context stored in tags to help you:

- manage your Cloud resources more cost effectively
- monitor and improve operational performance
- assess security and compliance

This guide *unifies* and *extends* the industry's disparate tagging recommendations into a comprehensive scheme covering: Development, Operations, Finance, Security, and Risk.

Teams can adopt this comprehensive tagging model in Cloud deployments today and accrue benefits with the next analysis of Cloud resources.

Who should read this paper

This paper is primarily intended for engineers, security specialists, systems architects, and IT professionals who are responsible for planning, executing, and operating application deployments on Cloud platforms.

These roles include the following common job descriptions:

- Engineers and technologists within organizations using Cloud platforms
- Architects and leaders who are responsible for driving the architecture efforts for their organizations
- Senior executives, business analysts, and colleagues across the business who have critical business objectives and requirements that need IT support, particularly in Finance, Security, and Risk Management
- Tool vendors who create specialized solutions to improve the efficiency, performance, and security of customers' Cloud deployments

Modeling Identity and Scope

Let's start with the context required to perform basic operational and cost management functions. The tags described in this section identify and scope resources to key organizational and process boundaries so you can answer questions like:

- Who owns this resource? What application does it belong to?
- Who should we call when the application is broken?
- Who should pay for this resource? Which applications are driving our costs?

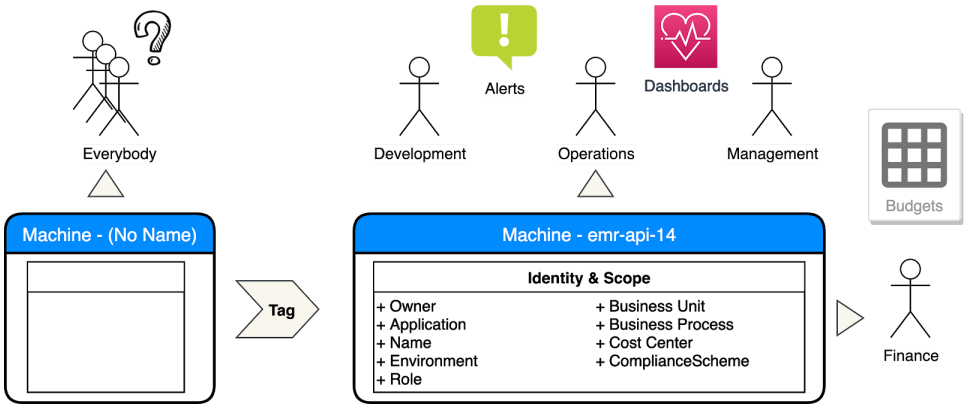

Figure 2: From No Context to Managing a Resource Tagged with Identity and Scope

The core set of tags used to identify and scope Cloud resources are:

- Owner: Identifies who is responsible for the resource
- Application: Identifies resources that support a specific application deployment
- Name: Identifies a resource with a name meaningful to people
- Environment: Identifies stage of Application delivery the resource belongs to
- Role: Describes the function of a resource within an Application's logical architecture
- Business Unit: Identifies the top-level organizational division that owns the resource
- Business Process: Identifies the high-level business process the resource supports
- Cost Center: Identifies the managerial accounting cost center for the resource
- Compliance Scheme: Identifies the regulatory compliance scheme the resource's configuration should conform to

These tag names and definitions are based on conventions used in practice and recommendations aggregated from a survey of five resource tagging standards in January

2020[1]. Some tags are only appropriate for large or regulated organizations. These tags help Cloud teams:

- Identify Application resources and their Owners, primarily for cost accounting and risk analysis
- Scope the resources involved in operating an Application or Environment, primarily so that people can solve performance and availability problems

Let's define each of these tags and illustrate usage with examples.

Owner

The `Owner` tag identifies who is responsible for the resource and is using that resource to perform a `Business Process`. The `Owner` tag is the single most important tag for a deployment.

The tag's value should be the name of an application team or department in the organization. Alternatively, the tag value should specify an email address, group chat, or other well-used communication channel the owning team monitors for notifications related to their Cloud deployments.

Example Values:

- Team or Department: Ecommerce, Data Science, Platform
- Communication Channel: ecommerce@org.com, #data-science-ops

Recommendations: The values of this tag should be easily verifiable and quickly resolvable to real people so that questions and alerts can be routed quickly. When specifying team names, valid values could be looked-up from an organizational metadata library. Communication channels can be sent a test message periodically to verify messages are still deliverable.

Application

The `Application` tag identifies resources that support a specific application deployment. This tag primarily assists monitoring and operational processes.

Example Values: ecommerce-frontend, order-processor, order-fulfillment, medical-records-api

Recommendations: Set this tag's value to the canonical technical name for the application used by the application's existing source control, delivery, and operational processes.

[1] Appendix - Surveyed Cloud Tagging Standards and Recommendations

Name

The `Name` tag identifies a resource with a name meaningful to the people that manage it. Every Cloud uniquely identifies each deployed resource, but many of these identifiers are not meaningful to people. For example, a virtual machine's id might be derived from its primary private IP address. When the Cloud's primary unique resource identifier is meaningful to people, the `Name` tag may be skipped to avoid repetition. A common example of this case are object storage buckets, whose names are used in DNS and HTTP. The `Name` tag assists development, monitoring and operational processes by naming resources so discussions refer to resources precisely, accurately, and naturally. `Name` tags are often displayed automatically in user interfaces.

Example Values:

- Dedicated application API VM: emr-api-dev-14
- Shared container cluster VM: dmz-shared-dev-02
- Database instance: emr-mysql-db-prod-02

Recommendations: Construct names from the minimal set of attributes people need to refer to describe the resource. Instances in a cluster may also need an automatically maintained number. The most useful attributes to include in a `Name` are short forms of the application name, environment, role, and security zone. Prefer a single, consistent naming format over a mix of formats incorporating different details. Describe important resource attributes with tags instead of trying to include everything in the name.

Environment

The `Environment` tag identifies which stage of `Application` delivery the resource belongs to and distinguishes between development, test, and production resources. This tag assists cost analysis, operations, and security management processes.

Example Values: dev, stage, prod

Recommendations: Validate or enforce use of the organization's well-known software delivery environment names using deployment automation and compliance tools. Application deployments that create a fresh environment for each change can extend the value with the version control change identifier, e.g. `dev-b7dd27e`.

Role

The `Role` tag describes the function of a particular resource *within* an `Application`'s logical architecture. This tag assists engineers in operating applications by helping them decompose and monitor the distinct components or tiers within an application. Engineers can incorporate `Role` information into monitoring dashboards and alert routing rules.

Example Values: load balancer, web server, application server, message broker, database, object store

Recommendations: Extract a standard list of roles from your reference architectures and existing deployments. Distinguish between roles where that distinction supports an important difference in the operational processes.

Each organization will need to decide what granularity of Role values is appropriate for their organization and operational processes. Let's illustrate this variable granularity with an example.

A common and potentially complicated Role to model is that of a database. Should the values for a database role be:

- Database
- Relational Database, Document Database, Key-Value Store
- RDBMS - MySQL, RDBMS - PostgreSQL
- RDBMS - Function A, RDBMS - Function B, DocDB - Function C

The answer depends on factors like what deployment options are supported by your organization, whether applications routinely use multiple database clusters, and how you would like to use this data in monitoring dashboards and cost reports.

Business Unit

The `BusinessUnit` tag identifies the top-level division of the organization that owns the resource or Cloud account.

Example Values: Consumer Retail, Enterprise Solutions, Manufacturing, Electronic Medical Records (EMR), Payments, Data Analysis

Recommendations: Small organizations often start with a single line of business — one business unit. In that case, the value would be the same for all resources or derived from the Cloud account which contains the resources. Model significant internal capabilities such as data warehouse and analysis functions as business units.

Business Process

The `Business Process` tag identifies the high-level business process or function the Cloud resource supports. Business processes are large in scope and recognizable by stakeholders across the organization and also externally. This tag helps managers determine input costs, monitor the health of supporting resources, and assess risks to delivery of that business process.

Example Values:

- Common: Marketing, Sales, Customer Support, Technology Delivery, Technology Operations, Finance, Human Resources
- Ecommerce: Shopping, Order Processing, Fulfillment
- Health: Record Management, Provider Integration

Recommendations: Many organizations will observe high overlap of values in the `Business Unit` and `Business Process` because they are organized according to process. However, this overlap will usually be incomplete. Some important business processes do not have a dedicated Vice President and staff high in the organizational hierarchy. Rather, those responsibilities are spread across the organization.

Technology Delivery and Operations are good examples of critical business processes where each Business Unit may have independent or loosely-coordinated implementations of those processes. An organization with Retail and Enterprise business units may have different teams and approaches for delivering and operating the technology to support those business units. Organizations will find it useful to answer questions around how much cost or risk is associated with Delivery and Operations in each of the business units, and in aggregate. This will help focus the organization's resources and efforts on the most important Business Processes.

Cost Center

The `Cost Center` tag identifies the managerial accounting cost center associated with a resource. This tag helps managers responsible for a business process account for and manage the costs of resources delivering a business process. The `Cost Center` tag does this by tying costs for each back to the organization's normal accounting processes. This is most useful when operating multiple applications in a single Cloud account.

Recommendations: Ask your `BusinessUnit`'s management and accounting team whether a `Cost Center` will be useful and at what granularity. They may suggest using only the higher-level Business Process instead. When adopting `Cost Center`, start with coarse grained, department-level cost centers and get more specific as needed.

Compliance Scheme

The `ComplianceScheme` tag identifies the regulatory compliance scheme to which the resource's configuration should conform. The `ComplianceScheme` tag can be used by configuration analysis tools to determine if the resource complies with the standard and inform the scope of an audit.

Example Values: HIPAA, PCI, SOC2, N/A

Recommendations: Organizations with regulated business processes should identify the supporting resources with the appropriate `ComplianceScheme`. Tag resources that are not covered by a regulatory framework with `ComplianceScheme=N/A` to indicate the resource is not within audit scope.

Takeaways: Identity and Scope
These nine tags identify and scope resources for the purposes of operating Cloud applications and managing their costs effectively. The next section describes how to model the intended security of the deployment's information assets.

Modeling Security

Once Cloud resources have been identified and scoped to particular applications, business processes, and environments, you can ask questions about the security of those information assets such as:

- Who should have access?
- Are the security access controls for this resource appropriate?
- Do too many people or applications have access to Confidential data?

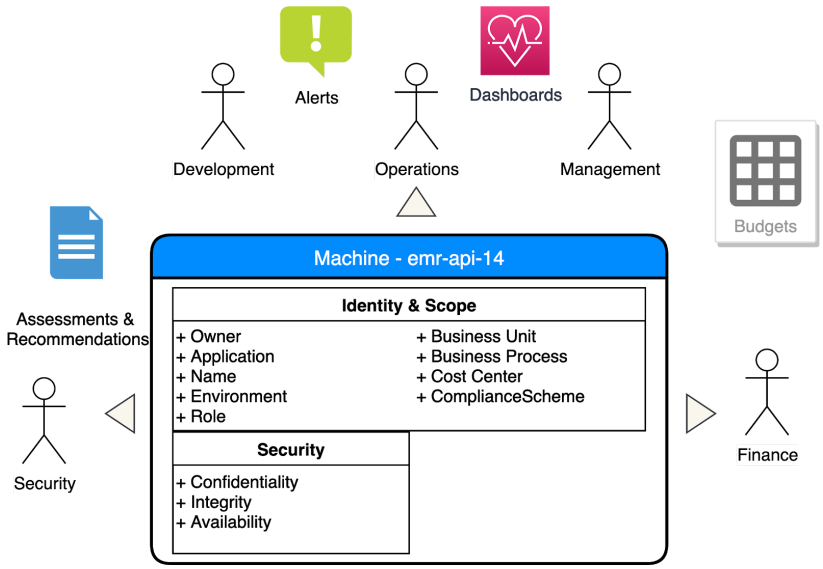

Figure 3: Managing Resource Tagged with Security Context

These are some of the questions people ask in security and risk assessments. The main inputs to the risk assessment process are:

- a risk model
- an assessment approach
- an analysis approach

- domain knowledge, what we will discuss in this paper
- current configuration

The risk assessment's domain knowledge inputs must describe the resources and environment being assessed in sufficient detail to analyze the information's security and risks.

This application security domain knowledge *is the **critical** descriptive information missing from most Cloud deployments*.

This information security context enables assessors and tools to determine relevance of vulnerabilities and configurations as they assess security, compliance, and risk.

Start by tagging resources in a Cloud deployment with stakeholders' *expectations* for the confidentiality, integrity and availability of information processed or stored by that resource. This aligns your Cloud deployments with mainstream information security and risk management models. Once this domain knowledge is available in tags, Cloud and Security teams can automate much of the recurring security and compliance analysis.

Let's define those security tags and their suggested values.

Confidentiality (or Data Classification)

The `Confidentiality (or Data Classification)` tag specifies stakeholders' *intended* level of confidentiality and uses of the data processed or stored by this resource, both inside and outside the organization.

Suggested Values: The suggested values for this tag come from the SANS Institute's Data Classifications:

- Public: Non-sensitive information available for external release
- Internal: Information that is generally available to employees and approved non-employees
- Confidential: Information that is sensitive within the company and is intended for use only by specified groups of employees
- Restricted: Information that is extremely sensitive and is intended for use only by named individuals within the company

Many organizations already have a data classification scheme (often SANS'), so determine if you can adopt it before defining a new standard.

Some of the most basic and important questions such as, "*should* this bucket be publicly accessible?" are answerable directly from a `Confidentiality` tag.

Buckets hosting public websites should be tagged with `Confidentiality=Public`. Buckets containing PHI or PCI data should be `Confidentiality=Confidential`. Classifying data as `Confidential` clearly indicates that the bucket should *also* have a bucket policy limiting access to the relevant applications and support personnel.

Describing the intended confidentiality of data processed by a resource is the most complicated of the three security attributes because confidentiality also implies a relationship with who and what is authorized to use the data. Consequently, engineers and assessors will likely be interested in additional context when evaluating confidentiality such as `Application`, `Business Process`, or `Owner`. Advanced security policies may use that additional context to limit access to only authorized identities.

Integrity

The `Integrity` tag specifies stakeholders' intended level of integrity for this data as required by the `Business Process`. This tag defines the extent to which the data must be guarded from improper modification or destruction in order for the `Business Process` to function acceptably. Integrity includes ensuring non-repudiation and authenticity of information.

Suggested Values: Measure integrity as the portion of records processed during a month that must maintain integrity before Business Process stakeholders declare an incident:

Integrity Level	Portion of Records with Integrity	Percentage of Records with Integrity
Two Nines	0.99	99%
Three Nines	0.999	99.9%
Four Nines	0.9999	99.99%
Five Nines	0.99999	99.999%
Six Nines	0.999999	99.9999%

Both the portion and percentage representations are understandable by people and tools.

Some people may wonder when you wouldn't specify 100% data integrity. The vast majority of application deployments can't achieve, don't need, or won't pay for 100% data integrity. For example, two use cases where people usually accept lower levels of integrity are application logs and monitoring data. By contrast, audit logs generally have strict integrity requirements and their integrity is often verifiable by cryptographic hashes.

A data resource with a strict six nines integrity requirement could be tagged with `Integrity=0.999999`. This communicates that access controls for writing and deleting data within that resource should be correspondingly strict.

Alternatively, you could define the inverse of maintaining integrity, and describe the maximum portion of processed records that may lose integrity in a month before an incident is declared. However, stating the measurement in the positive aligns the measurement direction with Availability, discussed next.

Availability

The `Availability` tag specifies stakeholders' desired portion of time or service requests that the resource should provide reliable and timely access in order for the `Business Process` to function acceptably, measured monthly in NINES of availability or allowed downtime per month.

Suggested Values: Measure availability as the portion of time or service requests that the resource should provide access, before `Business Process` stakeholders declare an incident:

Availability Level	Portion of Time or Requests Available	Allowed Downtime (monthly)
Two Nines	0.99	7.3 hours
Three Nines	0.999	10.08 minutes
Three and a Half Nines	0.9995	5.04 minutes
Four Nines	0.9999	1.01 minutes
Five Nines	0.99999	6.05 seconds

Here are some examples of applying this definition to resources:

First, a stateless load balancer or compute cluster must be deployed across three availability zones to achieve 0.9995 or better availability and should be tagged with `Availability=0.9995`.

Second, a web application that wants to achieve 99.95% availability and is using an RDBMS must handle a database failover as a complete system within 5 minutes. This means the database cluster services probably need close to four nines availability. If the DB cluster is tagged with `Availability=0.9999`, then operators and tools know they should expect:

- a warm replica running in another availability zone, because launching a new instance takes ~10 minutes
- configurations that safely and automatically promote a replica to leader within 3-4 minutes of a failure event

This leaves 1-2 minutes for the application database drivers to detect that failover and switch to the new leader unless the system provides a network endpoint that handles this transparently.

Takeaways: Security
This information security context provides the critical domain knowledge about how the organization *intends* to manage the information processed by that resource. This context can be used by people and tools to analyze and assess risks to the information security, availability, cost of downtime, and more.

When you apply these Security tags consistently, you should be able to assess and address the fundamental information security of resources quickly. First, compliance and basic security configuration checks are simpler to automate and scale. Second, conversations with owners about resource configurations will start from a much more informed place using standardized language.

Next, we'll show how to use the Identity, Scope, and Security context we have gathered so far to model the *loss* of information security and discuss the risk in terms everyone can understand: money.

Modeling Risk

The NIST 800-30 Guide to Performing Risk Assessments defines Information Security risks as those risks "that arise from the loss of confidentiality, integrity, or availability of information or information systems and reflect the potential adverse impacts to organizational operations, organizational assets, individuals, other organizations, and the Nation."

Risk assessments should help people understand a given Cloud deployment's information security risks and help leaders *make better decisions* when managing those risks.

Many people assess their information security risks broadly and infrequently. They may estimate the number of important data records and impacts of losing records annually.

Once you describe the security context of your (most critical) information assets' intended Confidentiality, Integrity, and Availability, you have the foundation to describe and then analyze risk in a fine-grained, scalable, repeatable way. The risk context modeled in this section will help you answer questions like:

- How much risk does our Cloud deployment have and where is that risk concentrated?
- Which security improvements reduce our risk the most?

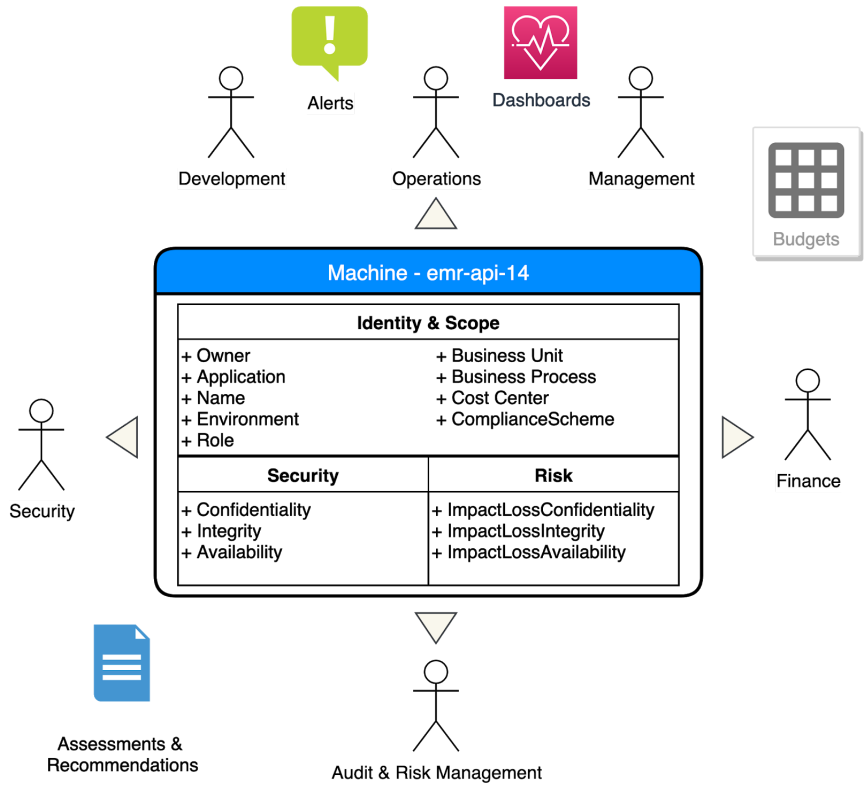

Figure 4: Managing a Resource with Risk Context

Record the most important Risk context of your information assets in terms of the impact of the *loss* of those information security attributes. Tag resources that process or store information with:

- `ImpactLossConfidentiality`
- `ImpactLossIntegrity`
- `ImpactLossAvailability`

The risk assessment process will use this domain knowledge of potential impacts. Let's quickly introduce the risk assessment process now.

Assessing Risk

Within a risk assessment process you first identify threats to information security such as:

- An external adversary attacks an application to gain network access and abuse the application's credentials to extract data from an RDBMS or Object Store like S3
- An engineer accidentally pushes infrastructure code that an automated delivery system dutifully applies to destroy a production database.

Those threats identify two of the biggest problems on the minds of Cloud, DevOps, Security, and Risk Management professionals today: data breaches and accidental data destruction.

After identifying threats to your information assets, estimate the **likelihood** those threats will materialize and the **impact** when they do. That likelihood and impact information plugs into the classic risk calculation function:

```
risk = (likelihood_confidentiality_loss * impact_confidentiality_loss)
       + (likelihood_integrity_loss * impact_integrity_loss)
       + (likelihood_availability_loss * impact_availability_loss)
```

Initially, this might feel unnatural as you try to put some definition to these new dimensions. You will get better with guidance and practice. Keep in mind that the goal here is to add context that helps you make *better* decisions, not perfect ones.

When someone asks you what the biggest risks to your information security assets are, you can say something like, "we think these are the biggest risks, and this is why" in a structured way instead of "no comment" or hand-waving through an ad-hoc explanation.

Impact

Start by estimating the impact of a loss of information confidentiality, integrity, or availability of resources in your Cloud deployment for threats such as a data breach. In general, you should assess the impact on information processed or stored by a resource, as opposed to assessing the impact on the information's 'container' resource such as a compute instance or storage bucket. Resources are generally fungible and easily replaceable, especially in the Cloud, so replacement or repair costs are often negligible.

This guide will describe two ways you can estimate the impact: qualitatively and quantitatively. K9 Security recommends working towards a quantitative approach, even if you start with a qualitative one.

As a preview, resources will be tagged with impacts like:

- Qualitative: `ImpactLossConfidentiality=VeryHigh`
- Quantitative: `ImpactLossConfidentiality=[1000, 1.0E+06]`

Let's explore qualitative and quantitative impact assessment methods now.

Qualitative

A qualitative impact assessment uses qualitative categories for values like Very High, High, Moderate, Low, Very Low. NIST 800-30 describes these values in detail. A qualitative impact assessment is a good starting point for conversations about risk and a coarse-grained risk analysis.

You could classify impact qualitatively in a tag like:

- `ImpactLossConfidentiality=VeryHigh`
- `ImpactLossIntegrity=Moderate`
- `ImpactLossAvailability=Low`

In the early stages of a risk management program, this should be enough information to identify obviously large problems and prioritize response efforts.

For example, you may suspect there are some big data protection problems out there and you have a limited set of resources or time to address them. In this case, filtering the impacts to confidentiality or integrity those with a value of `VeryHigh` or `High` and estimating the probability of those events occurring in your head might be good enough for the first pass.

However, qualitative values are often more difficult for leaders to use when making budget allocation decisions and prioritizing efforts over time. Challenges will surface as your risk management program matures.

Here are some questions that have difficult answers when using qualitative impact scales:

- How do I multiply a `High` impact and a probability to get an expected annual loss for, e.g. Confidentiality? How do I sum the individual risk components for an application? How do I do that consistently?
- Should I invest $1M to address a '`High`' risk? How does that compare to this other opportunity where $1M is expected to return $10M-$15M?
- What are my department or organization's expected information security related losses this year? Do I have enough Cybersecurity insurance?

The same problems exist to some degree with a semi-quantitative analysis that ranks loss impact on an ordinal scale of, e.g. 0-100.

Both qualitative and semi-quantitative analysis approaches are a good way to identify the subset of assets to focus quantitative analysis efforts on. Let's look at a quantitative impact assessment model now.

Quantitative

Quantitative analysis describes impacts in terms that make sense outside of your team: money.

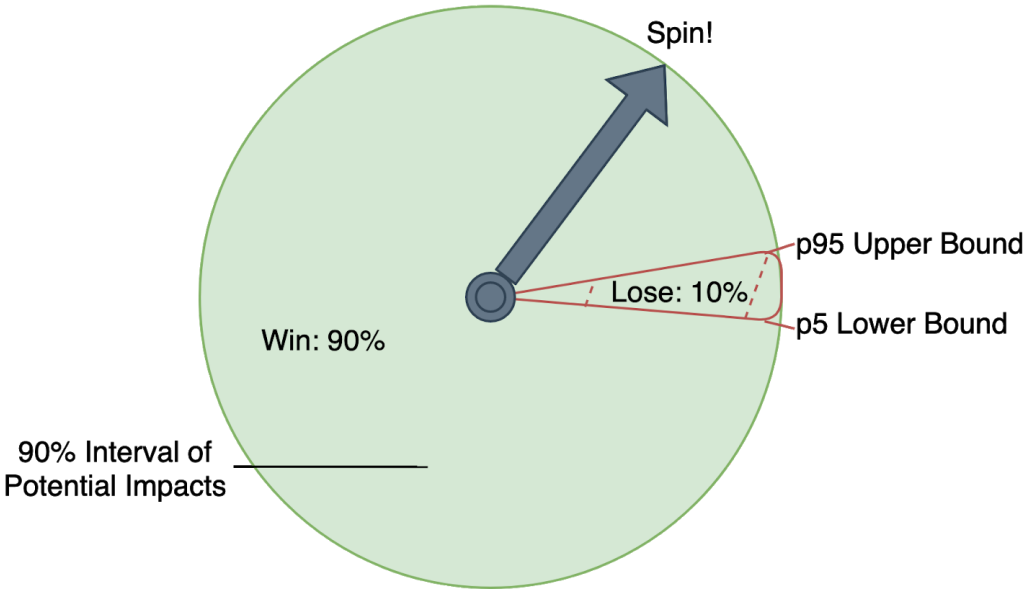

Figure 5: The Equivalent Bet Test aka The Loss Interval Clock [2]

Quantify a threat's impact using a confidence interval that covers the range of potential monetary losses you expect to incur from an event such as a data breach 90% of the time. The interval starts with the 5th-percentile of expected loss and ends at the 95th percentile, covering the likely 'best' and 'worst' case outcomes. This approach is used in many risk analysis methods, including FAIR. How to Measure Anything in Cybersecurity Risk (Hubbard) describes how to do this consistently with low effort.

[2] https://www.tonym-v.com/blog/2019/10/2/improve-your-estimations-with-the-equivalent-bet-test

For example, suppose we expect loss of confidentiality for the records in the production user database (think password hashes and PII) to cost:

- at least $1,000 if it leaks internally because we have to fix the problem and may need to clean up internal services like log storage and analysis systems
- at most $100,000 if the information is exfiltrated by an attacker because we have to notify our customers, provide some identity theft monitoring, and engage an incident response team to help us communicate and manage the impact on the public

This interval won't be perfect and it doesn't need to be. It is a reasonable estimate of the impact of an event based on the context that you know as a technology professional and understanding of your stakeholders.

When you share this impact interval, you'll likely end up in a conversation about the boundaries and shape of the distribution of loss. This is great because it means you've connected with that colleague or decision maker on terms they understand. Use that discussion to *update the estimate with that new information* and *move forward together*.

Once you have an impact interval, record this information in tags for each of the core Information Security attributes:

- `ImpactLossConfidentiality`
- `ImpactLossIntegrity`
- `ImpactLossAvailability`

Returning to the user database example, the impact of a loss of confidentiality would be described by:

```
ImpactLossConfidentiality=[1000, 1.0E+06]
```

See the Appendix for a detailed example of calculating the impact for availability incidents for a three tier web application. The availability loss estimate for that example is described as:

```
ImpactLossAvailability=[250, 1.9E+04]
```

The impact tag's value formats the loss range as a closed interval that begins with the lower bound of impact, ends with the upper bound, and contains numbers with two digits of precision in scientific notation. This format is understandable by people and tools like Excel, Google Sheets, Python, and Golang. You could also add currency units, e.g. `250USD`, though this will complicate parsing.

Likelihood

To complete the risk modeling, we need to estimate the likelihood or probability that an incident will occur within a particular timeframe.

K9 Security recommends managing likelihood information outside of the tagging scheme. The probability distributions used to model the frequency of events matters a lot and varies significantly across the attributes of Confidentiality, Integrity, and Availability for common threats. However, these probabilities are not particularly application or resource specific so there is little value in modeling likelihood information on each resource.

That said, you could create a parallel set of tags like `FrequencyLossAvailability=3`. Consider whether it is better to maintain this information inside your risk modeling and analysis tools or resource tags. It is probably easier to change and perform what-if analysis in the risk analysis tool than resource tags.

Fix the threat probability timeframe to one year. An annual time horizon aligns closely to the cadence at which many risk management programs operate and demonstrate investment benefits.

Takeaways: Risk

This risk context provides critical domain knowledge about how the organization will likely be impacted by information security risks. This context can be used by people and tools to understand and manage risks at the `Business Process`-level, which many organizations have little to no insight into. The risk context also enables discussion and prioritization of security control improvement efforts in a uniform way using standardized terms.

Summary

When you adopt the tags recommended by this guide, you will have the data required to:

- Identify resource owners
- Identify which resources support which applications and environments
- Understand and manage budgets for resource usage by Business Process, Application, Environment, and more
- Understand the intended major information security and availability attributes for data sources and application components
- Automate compliance, security, and risk analysis processes
- Understand where risk is concentrated in the Cloud deployment, enabling risk management at the Business Process level and individual Applications

Appendix - Surveyed Cloud Tagging Standards and Recommendations

The following Cloud tagging standards and recommendations were surveyed in January, 2020:

1. Amazon Web Services
2. Azure
3. Google Cloud Platform
4. Datadog
5. Apptio (Cloudability)

The survey identified the most common and useful tags as well as potential gaps describing security and risk attributes.

Appendix - Example: Impact Loss of Availability

Let's practice a bit using the availability attribute since many people are familiar with it.

Consider the impact of downtime to an ecommerce web application with an 99.95% availability requirement. Three and a half nines means 5 minutes of downtime is permitted monthly. Suppose you survey the previous year's incident reviews and find that there were 3 incidents:

1. unavailable for 17 minutes during peak hours
2. unavailable for 62 minutes during off-peak hours
3. unavailable for 29 minutes during peak hours on Cyber Monday

What's a reasonable loss interval? Suppose this ecommerce company makes $1,000 per hour off-peak, $5,000 per hour during non-holiday peaks, $15,000 per hour during holiday peaks. Based on the observed data, we would probably estimate that downtime events range in length from 15 minutes to maybe 75 minutes.
So a reasonable interval of impact for a single downtime incident could have:

- a lower bound of $250 (0.25 hours*$1000)
- an upper bound $18,750 (1.25 hours*15,000), which I'll round to two digits of precision:

  $19k

The loss impact interval could be described in a tag on the application's core dependencies such as database clusters as `ImpactLossAvailability=[250, 1.9E+04]`.

Now we have an idea of what one incident costs, somewhere between $250 and $19k per incident.