THE SCALE FACTORY

LESSONS LEARNED **FROM REVIEWING** 150 INFRASTRUCTURES_

JON TOPPER | @jtopper | he/him/his

$ whoami

- Founder/CEO/CTO The Scale Factory
- Working in hosting/infrastructure for 20 years

Infrastructure / AWS / DevOps

@jtopper

THE SCALE FACTORY

aws partner network

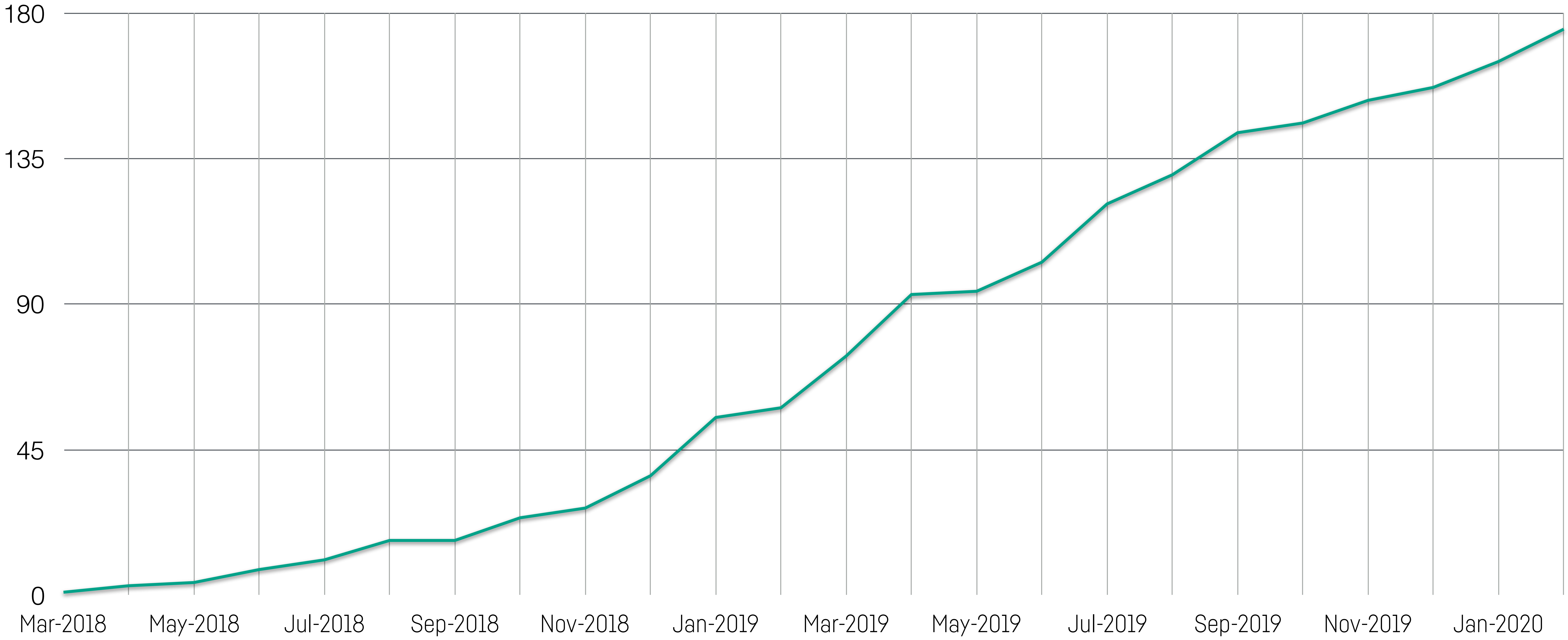
Advanced

Consulting Partner

Well Architected

@jtopper

REVIEWS
RUN_



AWS Well-Architected

@jtopper

TODAY'S **AGENDA_**

- What is Well-Architected?
- What is a Well-Architected Review?
- Common Review Findings

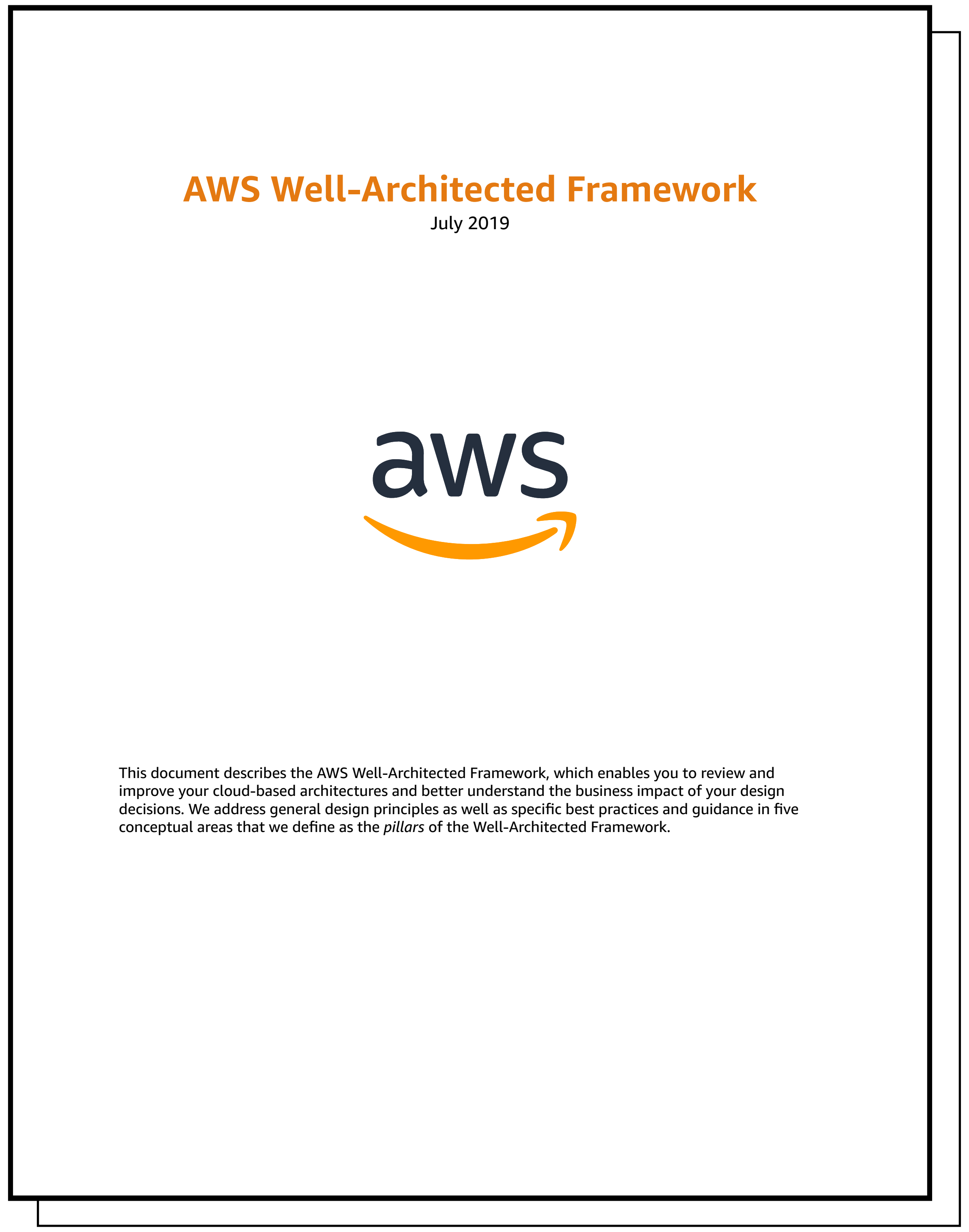
@jtopper

WHAT IS **WELL-ARCHITECTED?_**

WELL ARCHITECTED ORIGINS_

- Catalogue of emergent good practices
- Observed by AWS Field Solutions Architects
- Codified and shared
- Platform agnostic*

AWS Well-Architected

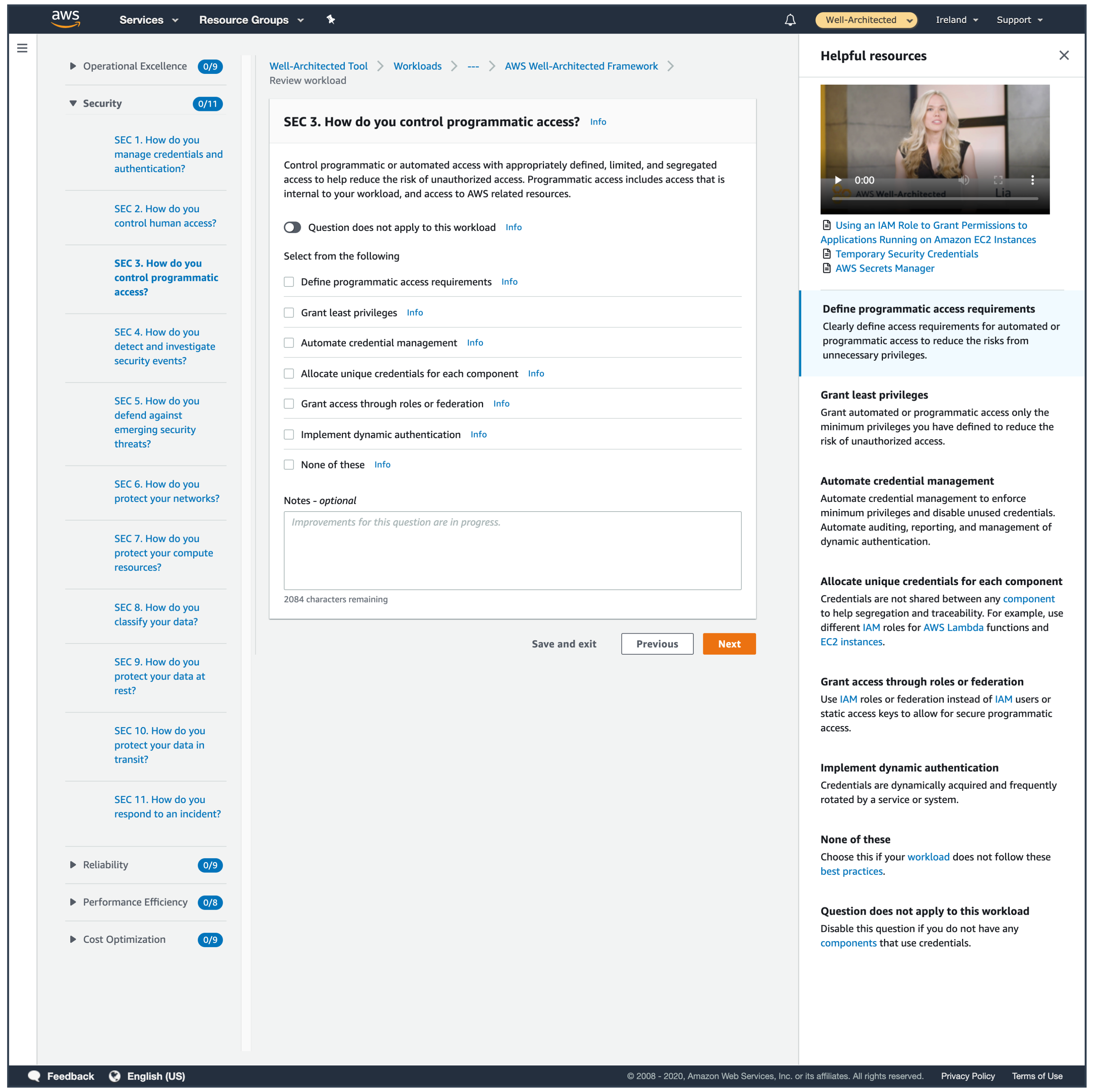


AWS Well-Architected Framework

July 2019

This document describes the AWS Well-Architected Framework, which enables you to review and improve your cloud-based architectures and better understand the business impact of your design decisions. We address general design principles as well as specific best practices and guidance in five conceptual areas that we define as the *pillars* of the Well-Architected Framework.

White Papers


Services
Resource Groups
Well-Architected
Ireland
Support
Operational Excellence 0/9
Security 0/11
SEC 1. How do you manage credentials and authentication?
SEC 2. How do you control human access?
SEC 3. How do you control programmatic access?
SEC 4. How do you detect and investigate security events?
SEC 5. How do you defend against emerging security threats?
SEC 6. How do you protect your networks?
SEC 7. How do you protect your compute resources?
SEC 8. How do you classify your data?
SEC 9. How do you protect your data at rest?
SEC 10. How do you protect your data in transit?
SEC 11. How do you respond to an incident?
Reliability 0/9
Performance Efficiency 0/8
Cost Optimization 0/9
Well-Architected Tool > Workloads > --- > AWS Well-Architected Framework >
Review workload
0:00
Save and exit
Previous
Next
Feedback
English (US)
Privacy Policy
Terms of Use

SEC 3. How do you control programmatic access? Info

Control programmatic or automated access with appropriately defined, limited, and segregated access to help reduce the risk of unauthorized access. Programmatic access includes access that is internal to your workload, and access to AWS related resources.

○ Question does not apply to this workload Info

Select from the following

- □ Define programmatic access requirements Info
- □ Grant least privileges Info
- □ Automate credential management Info
- □ Allocate unique credentials for each component Info
- □ Grant access through roles or federation Info
- □ Implement dynamic authentication Info
- □ None of these Info

Notes - *optional*

Improvements for this question are in progress.

2084 characters remaining

Helpful resources

- Using an IAM Role to Grant Permissions to Applications Running on Amazon EC2 Instances
- Temporary Security Credentials
- AWS Secrets Manager

Define programmatic access requirements
Clearly define access requirements for automated or programmatic access to reduce the risks from unnecessary privileges.

Grant least privileges
Grant automated or programmatic access only the minimum privileges you have defined to reduce the risk of unauthorized access.

Automate credential management
Automate credential management to enforce minimum privileges and disable unused credentials. Automate auditing, reporting, and management of dynamic authentication.

Allocate unique credentials for each component
Credentials are not shared between any component to help segregation and traceability. For example, use different IAM roles for AWS Lambda functions and EC2 instances.

Grant access through roles or federation
Use IAM roles or federation instead of IAM users or static access keys to allow for secure programmatic access.

Implement dynamic authentication
Credentials are dynamically acquired and frequently rotated by a service or system.

None of these
Choose this if your workload does not follow these best practices.

Question does not apply to this workload
Disable this question if you do not have any components that use credentials.

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Review Tool

AWS Well-Architected

Operational Excellence

Security

Reliability

Performance Efficiency

Cost Optimisation

AWS **Well-Architected**

Lenses

Serverless Applications

High Performance Computing

IoT (Internet of Things)

USING WELL-ARCHITECTED_

- Gap analysis / planning
- Teaching
- Team alignment

WHAT IS A **WELL-ARCHITECTED** REVIEW?_

WELL ARCHITECTED REVIEW_

- Foundational questions
- Up to 4 hours
- Qualitative

	Operational Excellence	Security	Reliability	Performance Efficiency	Cost Optimisation	
Well Architected Core	9	11	9	8	9	46
🔍 Serverless Applications	2	3	2	1	1	9
🔍 High Performance Computing	4	3	3	4	2	16
🔍 IoT (Internet of Things)	4	11	6	10	4	35

QUESTION
OPS 1_

How do you determine what your priorities are?

- Evaluate external customer needs □
- Evaluate internal customer needs □
- Evaluate compliance requirements □
- Evaluate threat landscape □
- Evaluate tradeoffs □
- Manage benefits and risks □
- None of these □

@jtopper

QUESTION

OPS 1_

How do you determine what your priorities are?

· Evaluate external customer needs	□	WA
· Evaluate internal customer needs	□	WA
· Evaluate compliance requirements	□	WA
· Evaluate threat landscape	□	NI
· Evaluate tradeoffs	□	NI
· Manage benefits and risks	□	NI
· None of these	□	CI

@jtopper

QUESTION

OPS 1_

High Risk

How do you determine what your priorities are?

· Evaluate external customer needs	□	WA
· Evaluate internal customer needs	□	WA
· Evaluate compliance requirements	□	WA
· Evaluate threat landscape	□	NI
· Evaluate tradeoffs	□	NI
· Manage benefits and risks	□	NI
· None of these	☑	CI

@jtopper

QUESTION

OPS 1_

Medium Risk

How do you determine what your priorities are?

Evaluate external customer needs	☑	WA
Evaluate internal customer needs	☑	WA
Evaluate compliance requirements	☑	WA
Evaluate threat landscape	□	NI
Evaluate tradeoffs	□	NI
Manage benefits and risks	□	NI
None of these	□	CI

@jtopper

QUESTION

OPS 1_

Medium Risk

How do you determine what your priorities are?

Evaluate external customer needs	☑	WA
Evaluate internal customer needs	☑	WA
Evaluate compliance requirements	☑	WA
Evaluate threat landscape	☑	NI
Evaluate tradeoffs	☑	NI
Manage benefits and risks	☐	NI
None of these	☐	CI

@jtopper

QUESTION

OPS 1_

Well Architected

How do you determine what your priorities are?

· Evaluate external customer needs	☑	WA
· Evaluate internal customer needs	☑	WA
· Evaluate compliance requirements	☑	WA
· Evaluate threat landscape	☑	NI
· Evaluate tradeoffs	☑	NI
· Manage benefits and risks	☑	NI
· None of these	□	CI

@jtopper

COMMON **REVIEW** FINDINGS_

THE GOOD_

QUESTION

OPS 1

Well Architected

77%

WA Rank: 1

How do you determine what your priorities are?

Evaluate external customer needs	WA	93%
Evaluate internal customer needs	WA	87%
Evaluate compliance requirements	WA	90%
Evaluate threat landscape	NI	85%
Evaluate tradeoffs	NI	89%
Manage benefits and risks	NI	89%
None of these	CI	0%

@jtopper

QUESTION
PERF 3_

Well Architected

70%

WA Rank: 2

How do you select your storage solution?

Understand storage characteristics and requirements	WA	84%
Evaluate available configuration options	NI	78%
Make decisions based on access patterns and metrics	NI	73%
None of these	CI	5%

@jtopper

QUESTION
REL 5_

Well Architected

63%

WA Rank: 3

How do you implement change?

Deploy changes in a planned manner	WA	83%
Deploy changes with automation	NI	67%
None of these	CI	6%

@jtopper

THE
BAD_

QUESTION
REL 9_

High Risk

79%

[87%]

HRI Rank: 1

How do you plan for disaster recovery?

Define recovery objectives for downtime and data loss	WA	33%
Use defined recovery strategies to meet the recovery objectives	WA	33%
Test disaster recovery implementation to validate the implementation	WA	25%
Manage configuration drift on all changes	NI	39%
Automate recovery	NI	16%
None of these	CI	31%

@jtopper

QUESTION
SEC 11_

High Risk

75%

(93%)

HRI Rank: 2

How do you respond to a [security] incident?

Identify key personnel and external resources	WA	51%
Identify tooling	WA	27%
Develop incident response plans	WA	39%
Automate containment capability	NI	0%
Identify forensic capabilities	NI	11%
Pre-provision access	NI	27%
Pre-deploy tools	NI	10%
Run game days	NI	3%
None of these	CI	35%

@jtopper

QUESTION
SEC 8_

High Risk

75%

(88%)

HRI Rank: 3

How do you classify your data?

Define data classification requirements	WA	61%
Define data protection controls	WA	39%
Implement data identification	WA	17%
Automate identification and classification	NI	4%
Identify the types of data	NI	59%
None of these	CI	23%

@jtopper

QUESTION

COST 9_

High Risk

71%

(79%)

HRI Rank: 4

How do you evaluate new services?

Establish a cost optimisation function	WA	34%
Develop a workload review process	WA	26%
Review and implement services in an unplanned way	NI	84%
Review and analyse this workload regularly	NI	43%
Keep up to date with new service releases	NI	63%
None of these	CI	1%

@jtopper

How do you test resilience?

QUESTION

REL 8_

High Risk

67%

[92%]

HRI Rank: 5

Use playbooks for unanticipated failures	WA	25%
Conduct root cause analysis and share results	WA	73%
Inject failures to test resiliency	NI	6%
Conduct game days regularly	NI	0%
None of these	CI	16%

@jtopper

THE
NOTABLE_

@jtopper

QUESTION

OPS 3_

Well Architected

14%

WA Rank: 23

How do you reduce defects, ease remediation, and improve flow into production?

Use version control	WA	90%
Test and validate changes	WA	87%
Use config management systems	NI	78%
Use build/deploy systems	NI	82%
Perform patch management	NI	37%
Share design standards	NI	57%
Implement practices to improve code quality	NI	83%
Use multiple environments	NI	81%
Make frequent, small, reversible changes	NI	63%
Fully automate integration and deployment	NI	52%
None of these	CI	3%

@jtopper

QUESTION
OPS 6_

Well Architected

46%

WA Rank: 21

How do you understand the health of your workload?

Identify key performance indicators	WA	53%
Define workload metrics	WA	62%
Collect and analyse workload metrics	WA	72%
Establish workload metric baselines	NI	51%
Learn expected patterns of activity for workload	NI	54%
Alert when workload outcomes are at risk	NI	40%
Alert when workload anomalies are detected	NI	34%
Validate the achievement of outcomes and the effectiveness of KPIs and metrics	NI	37%
None of these	CI	14%

@jtopper

QUESTION
SEC 2_

High Risk

47%

[88%]

HRI Rank: 20

How do you control human access?

Define human access requirements	WA	70%
Grant least privileges	WA	58%
Allocate unique credentials per person	WA	90%
Manage credentials based on lifecycle	NI	70%
Automate credential management	NI	13%
Grant access through roles or federation	NI	62%
None of these	CI	3%

@jtopper

QUESTION

SEC 3_

High Risk

57%

[89%]

HRI Rank: 15

How do you control programmatic access?

Define programmatic access requirements	WA	40%
Grant least privileges	WA	70%
Automate credential management	NI	24%
Allocate unique credentials per component	NI	68%
Grant access through roles or federation	NI	58%
Implement dynamic authentication	NI	22%
None of these	CI	13%

@jtopper

MAJOR
THEMES_

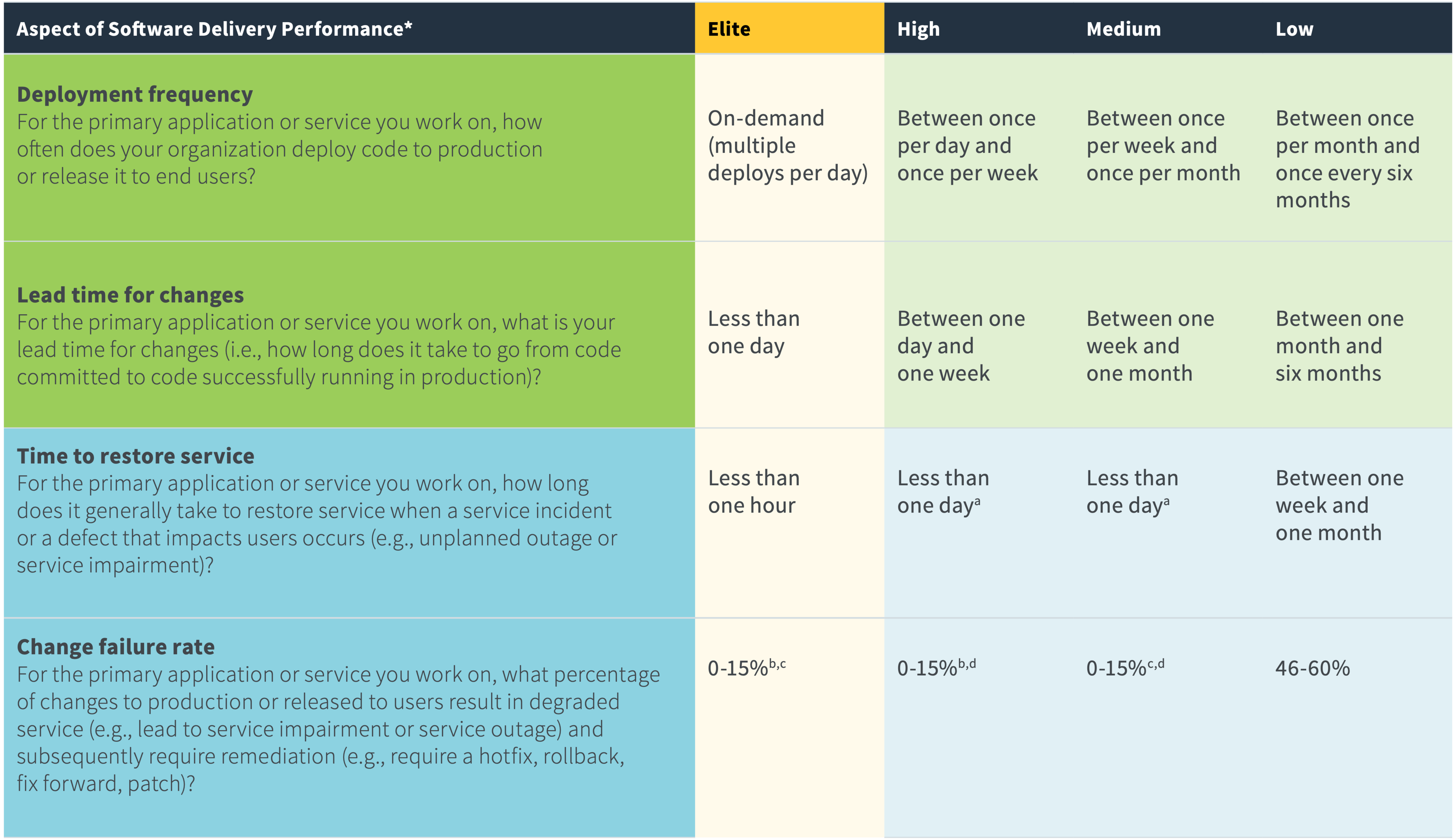
@jtopper

TEAMS ARE OK AT CHOOSING CORRECT **SERVICES**_

- Database choices match workload
- Storage choices match workload
- Compute choices sometimes not right-sized.

TEAMS ARE OK AT MAKING SOFTWARE **CHANGES**_

- Automation tools are being used
- Full CD remains out of reach
- Change batch sizes need to be smaller

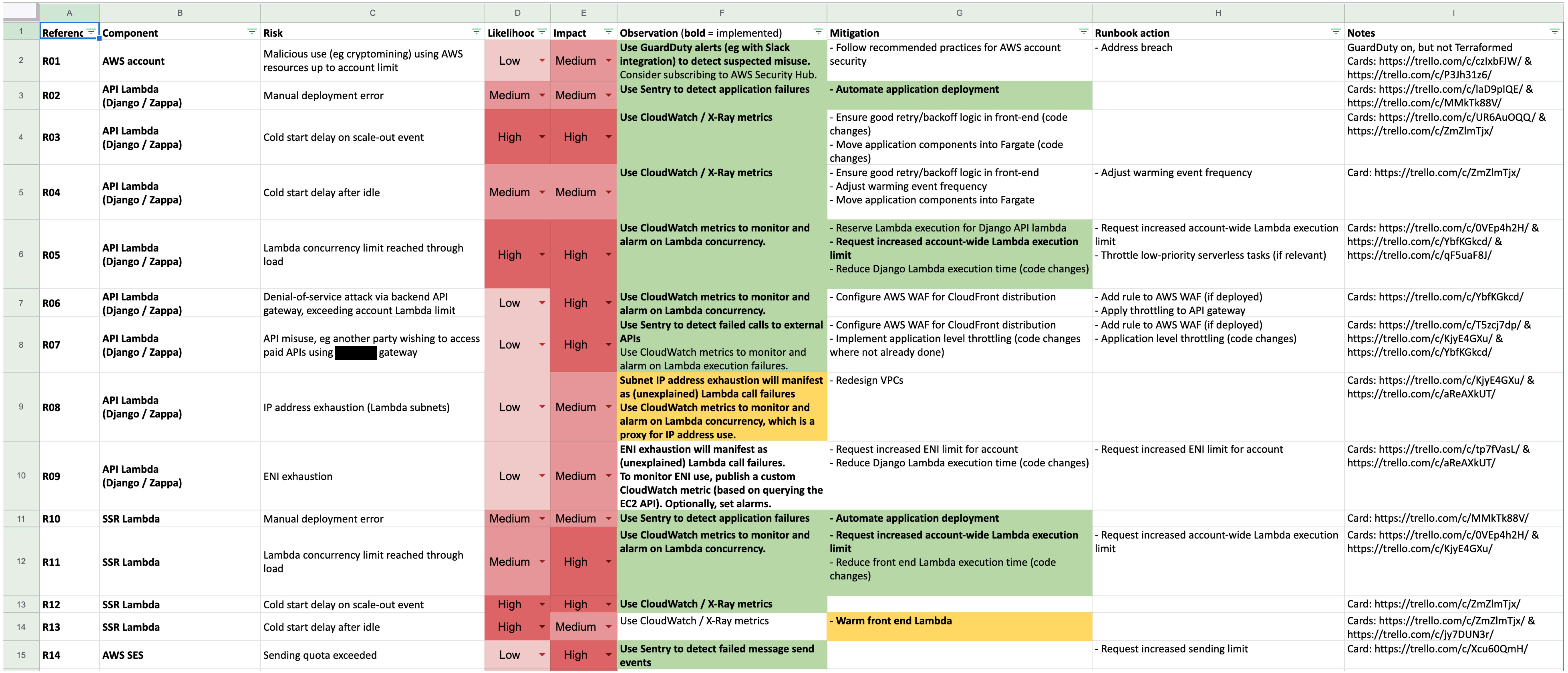


Aspect of Software Delivery Performance*	Elite	High	Medium	Low
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per day and once per week	Between once per week and once per month	Between once per month and once every six months
Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one day	Between one day and one week	Between one week and one month	Between one month and six months
Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour	Less than one day[a]	Less than one day[a]	Between one week and one month
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0-15%[b,c]	0-15%[b,d]	0-15%[c,d]	46-60%

https://services.google.com/fh/files/misc/state-of-devops-2019.pdf

TEAM ARE BAD AT THINKING ABOUT FAILURE MODES_

- Not considering business requirements
- No risk analysis of failure modes
- Poor documentation
- Almost no attempt to rehearse outages



	A	B	C	D	E	F	G	H	I
1	**Referenc**	**Component**	**Risk**	**Likelihooc**	**Impact**	**Observation (bold** = implemented)	**Mitigation**	**Runbook action**	**Notes**
2	**R01**	**AWS account**	Malicious use (eg cryptomining) using AWS resources up to account limit	Low	Medium	**Use GuardDuty alerts (eg with Slack integration) to detect suspected misuse.** Consider subscribing to AWS Security Hub.	- Follow recommended practices for AWS account security	- Address breach	GuardDuty on, but not Terraformed Cards: https://trello.com/c/czIxbFJW/ & https://trello.com/c/P3Jh31z6/
3	**R02**	**API Lambda (Django / Zappa)**	Manual deployment error	Medium	Medium	**Use Sentry to detect application failures**	**- Automate application deployment**		Cards: https://trello.com/c/laD9plQE/ & https://trello.com/c/MMkTk88V/
4	**R03**	**API Lambda (Django / Zappa)**	Cold start delay on scale-out event	High	High	**Use CloudWatch / X-Ray metrics**	- Ensure good retry/backoff logic in front-end (code changes) - Move application components into Fargate (code changes)		Cards: https://trello.com/c/UR6AuOQQ/ & https://trello.com/c/ZmZlmTjx/
5	**R04**	**API Lambda (Django / Zappa)**	Cold start delay after idle	Medium	Medium	**Use CloudWatch / X-Ray metrics**	- Ensure good retry/backoff logic in front-end - Adjust warming event frequency - Move application components into Fargate	- Adjust warming event frequency	Card: https://trello.com/c/ZmZlmTjx/
6	**R05**	**API Lambda (Django / Zappa)**	Lambda concurrency limit reached through load	High	High	**Use CloudWatch metrics to monitor and alarm on Lambda concurrency.**	- Reserve Lambda execution for Django API lambda **- Request increased account-wide Lambda execution limit** - Reduce Django Lambda execution time (code changes)	- Request increased account-wide Lambda execution limit - Throttle low-priority serverless tasks (if relevant)	Cards: https://trello.com/c/0VEp4h2H/ & https://trello.com/c/YbfKGkcd/ & https://trello.com/c/qF5uaF8J/
7	**R06**	**API Lambda (Django / Zappa)**	Denial-of-service attack via backend API gateway, exceeding account Lambda limit	Low	High	**Use CloudWatch metrics to monitor and alarm on Lambda concurrency.**	- Configure AWS WAF for CloudFront distribution	- Add rule to AWS WAF (if deployed) - Apply throttling to API gateway	Cards: https://trello.com/c/YbfKGkcd/
8	**R07**	**API Lambda (Django / Zappa)**	API misuse, eg another party wishing to access paid APIs using [redacted] gateway	Low	High	**Use Sentry to detect failed calls to external APIs** Use CloudWatch metrics to monitor and alarm on Lambda execution failures.	- Configure AWS WAF for CloudFront distribution - Implement application level throttling (code changes where not already done)	- Add rule to AWS WAF (if deployed) - Application level throttling (code changes)	Cards: https://trello.com/c/T5zcj7dp/ & https://trello.com/c/KjyE4GXu/ & https://trello.com/c/YbfKGkcd/
9	**R08**	**API Lambda (Django / Zappa)**	IP address exhaustion (Lambda subnets)	Low	Medium	**Subnet IP address exhaustion will manifest as (unexplained) Lambda call failures** **Use CloudWatch metrics to monitor and alarm on Lambda concurrency, which is a proxy for IP address use.**	- Redesign VPCs		Cards: https://trello.com/c/KjyE4GXu/ & https://trello.com/c/aReAXkUT/
10	**R09**	**API Lambda (Django / Zappa)**	ENI exhaustion	Low	Medium	**ENI exhaustion will manifest as (unexplained) Lambda call failures.** **To monitor ENI use, publish a custom CloudWatch metric (based on querying the EC2 API). Optionally, set alarms.**	- Request increased ENI limit for account - Reduce Django Lambda execution time (code changes)	- Request increased ENI limit for account	Cards: https://trello.com/c/tp7fVasL/ & https://trello.com/c/aReAXkUT/
11	**R10**	**SSR Lambda**	Manual deployment error	Medium	Medium	**Use Sentry to detect application failures**	**- Automate application deployment**		Card: https://trello.com/c/MMkTk88V/
12	**R11**	**SSR Lambda**	Lambda concurrency limit reached through load	Medium	High	**Use CloudWatch metrics to monitor and alarm on Lambda concurrency.**	**- Request increased account-wide Lambda execution limit** - Reduce front end Lambda execution time (code changes)	- Request increased account-wide Lambda execution limit	Cards: https://trello.com/c/0VEp4h2H/ & https://trello.com/c/KjyE4GXu/
13	**R12**	**SSR Lambda**	Cold start delay on scale-out event	High	High	**Use CloudWatch / X-Ray metrics**			Card: https://trello.com/c/ZmZlmTjx/
14	**R13**	**SSR Lambda**	Cold start delay after idle	High	Medium	Use CloudWatch / X-Ray metrics	**- Warm front end Lambda**		Cards: https://trello.com/c/ZmZlmTjx/ & https://trello.com/c/jy7DUN3r/
15	**R14**	**AWS SES**	Sending quota exceeded	Low	High	**Use Sentry to detect failed message send events**		- Request increased sending limit	Card: https://trello.com/c/Xcu60QmH/

@jtopper

TEAMS ARE BAD AT MONITORING FOR FAILURE MODES_

- Monitoring happening
- Data not used for much
- Tracing almost non-existent

TEAMS NEED TO DO BETTER AT SECURITY_

- Poor hygiene around patching
- Limited data classification
- Mediocre human access control
- Bad programmatic access control
- Low adoption of security monitoring tools

TOP BREACH **CAUSES_**

- Using components with known vulnerabilities
- Security misconfiguration
- Injection
- Weak auth / session management
- Missing function access control

@jtopper

https://snyk.io/blog/owasp-top-10-breaches/

EVERYONE IS BETTER AT
BUILDING PLATFORMS
THAN THEY ARE AT
SECURING OR RUNNING THEM_

WHAT NEXT?_

- Read the white papers:

  https://aws.amazon.com/architecture/well-architected/

- Run your own review(s)

  https://aws.amazon.com/well-architected-tool/

- Consider engaging an AWS Well-Architected partner

  https://scalefactory.com/services/well-architected/

  (funding available)

KEEP IN TOUCH_

http://www.scalefactory.com/

https://github.com/scalefactory

@scalefactory

jon@scalefactory.com