

Serverless : 8 tendances qui caractérisent son adoption

L'usage d'AWS Lambda peut-il refléter la réalité du *serverless* ?

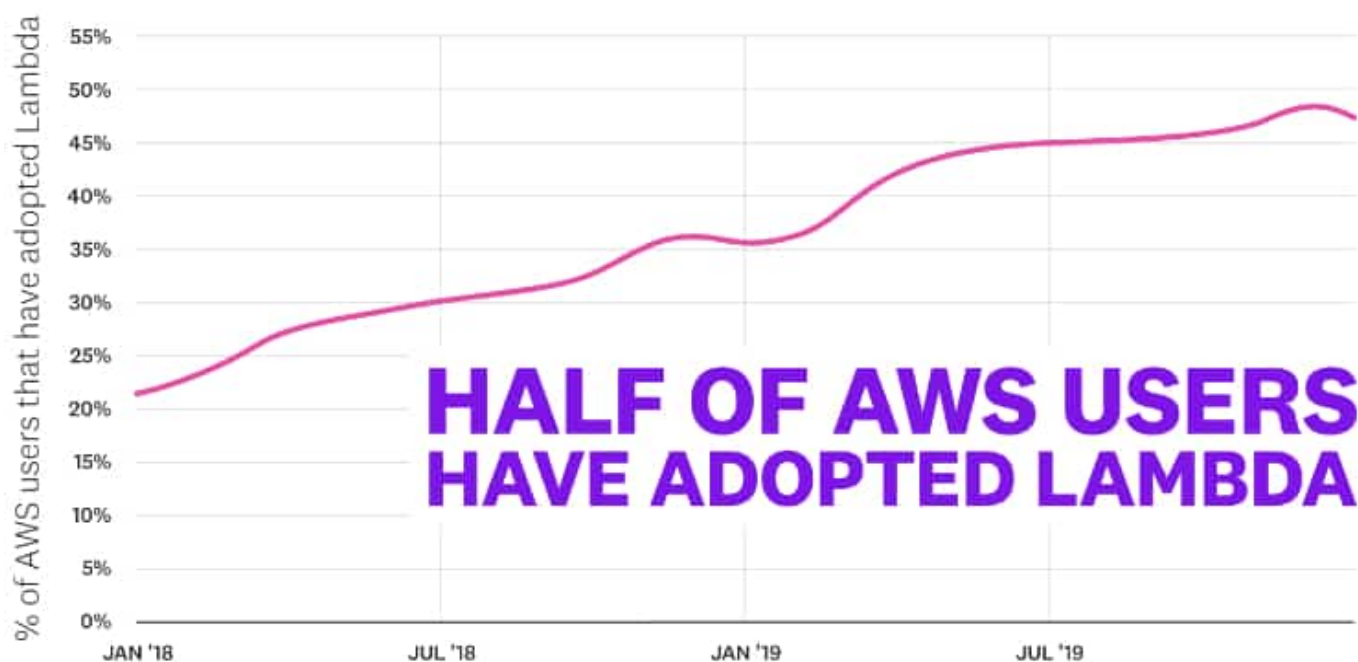
Datadog a pris ce parti pour constituer son [premier rapport sur le sujet](#).

L'éditeur américain, à l'origine d'outils de monitoring des infrastructures et des applications, s'est appuyé sur les données d'utilisation de « milliers » de ses clients. « *Un échantillon large, mais pas parfaitement représentatif de l'ensemble du marché mondial* », prend-il la peine de reconnaître.

Dans sa nomenclature, une organisation est considérée comme :

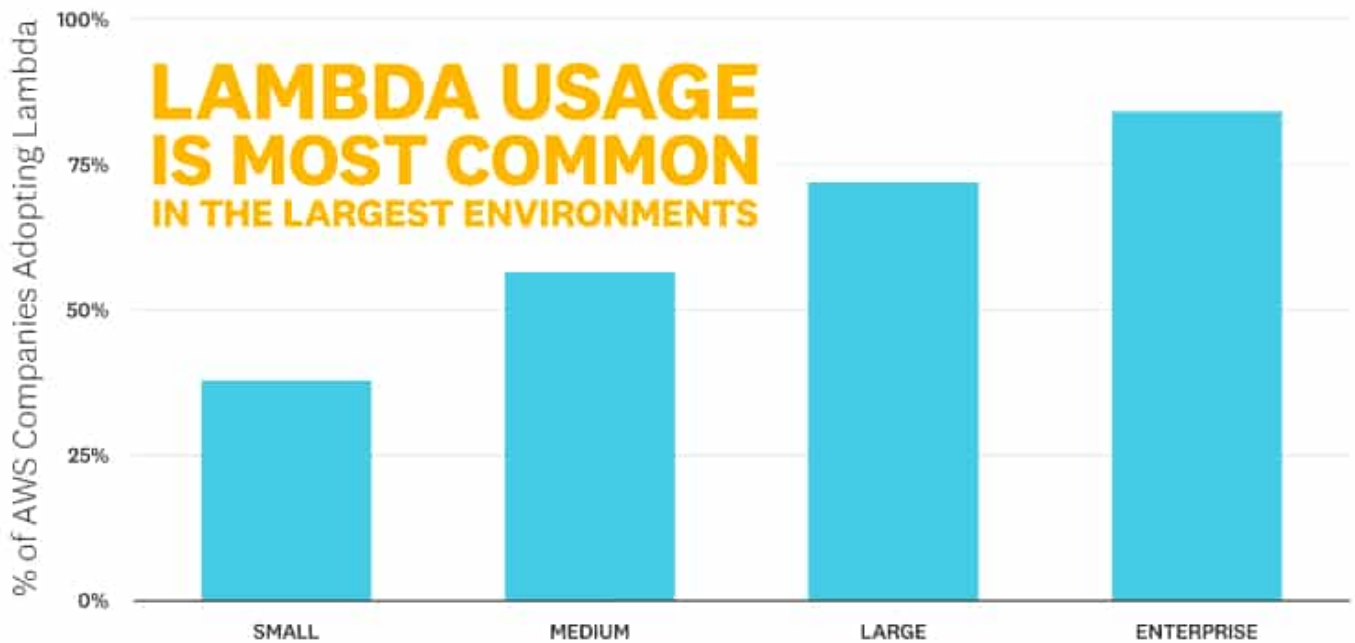
- utilisatrice d'AWS si elle a géré, au cours d'un mois donné, au moins 5 fonctions Lambda ou au moins 5 instances EC2 ;
- adoptrice de Lambda si elle a exécuté, au cours d'un mois donné, au moins 5 fonctions Lambda.

Sur cette base, le taux d'adoption de Lambda atteignait, début 2020, 47 % chez les organisations utilisatrices d'AWS. Il était d'environ 35 % un an plus tôt et 20 % deux ans plus tôt.



Les données de Datadog dénotent une corrélation entre la taille – estimée – de l'infrastructure et l'usage de Lambda. Celui-ci est présent dans environ 40 % des « petits » environnements, contre près de 80 % des « très gros ».

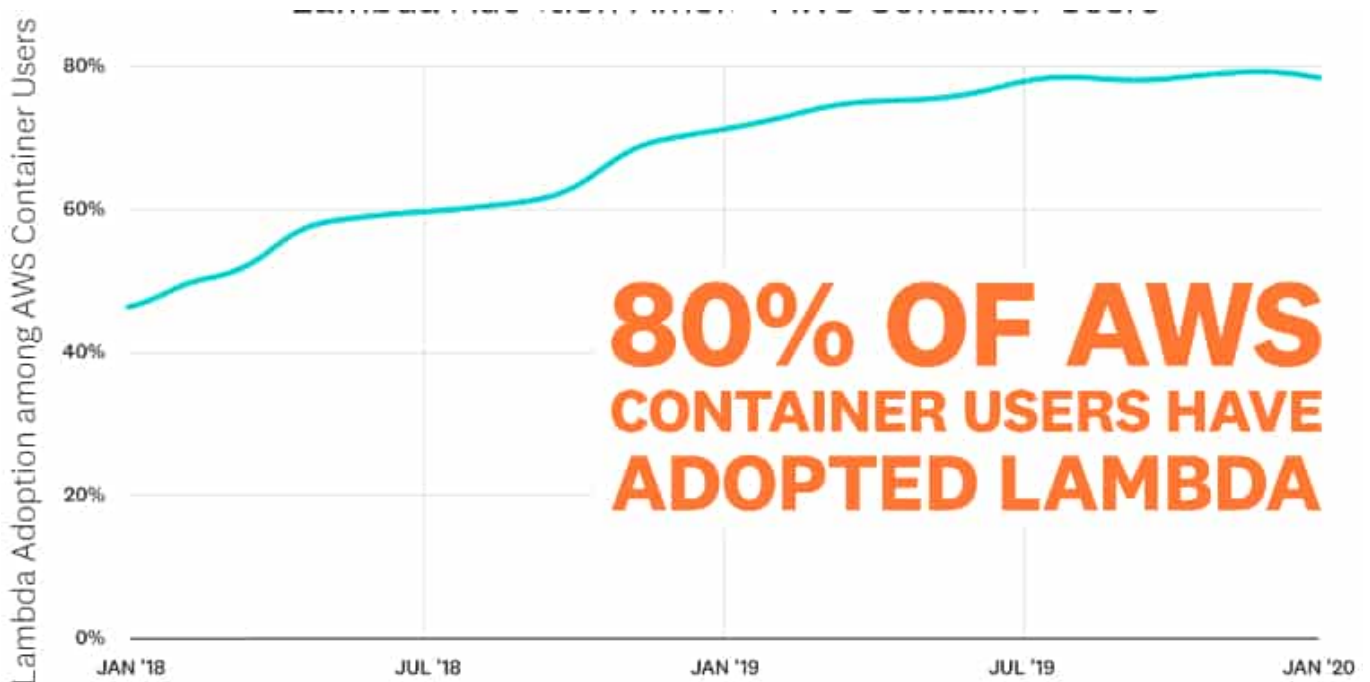
Lambda Usage by Environment Size



Python en pole

Les deux ne sont pas nécessairement liés, mais près de 80 % des organisations exécutant des conteneurs ont adopté Lambda.

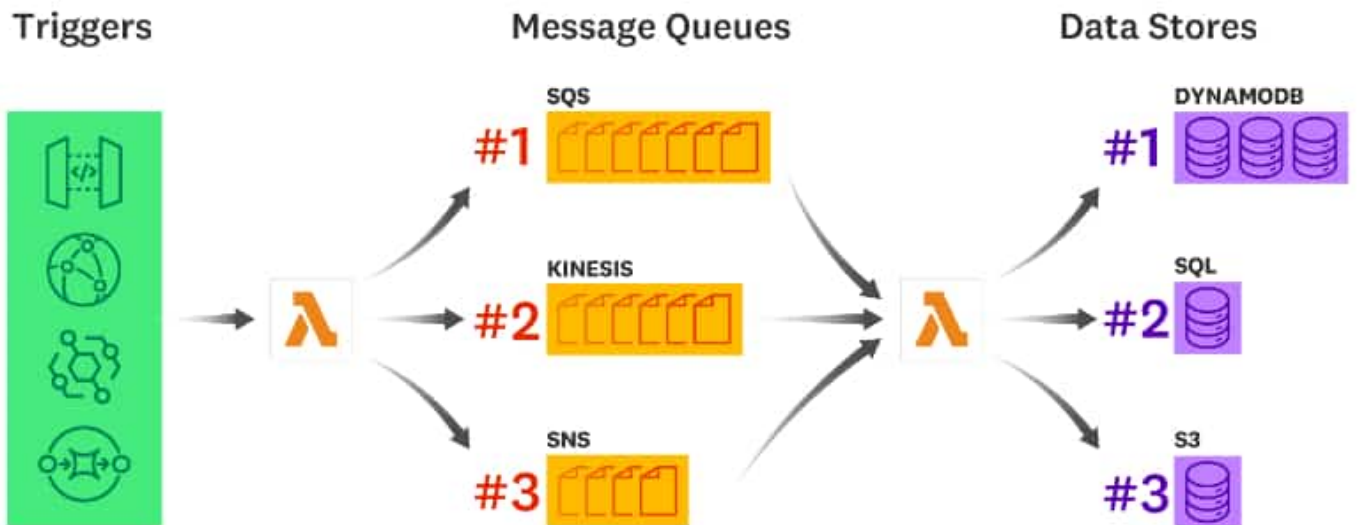
La proportion était inférieure à 50 % en janvier 2018. Elle dépassait les 70 % en janvier 2019.



Pour accompagner Lambda d'une couche de stockage persistant, DynamoDB est le premier choix. Viennent ensuite SQL (en instances RDS ou non) et Amazon S3.

Pour la gestion des files d'attente de messages, SQS est privilégié à deux autres services Amazon : Kinesis et SNS.

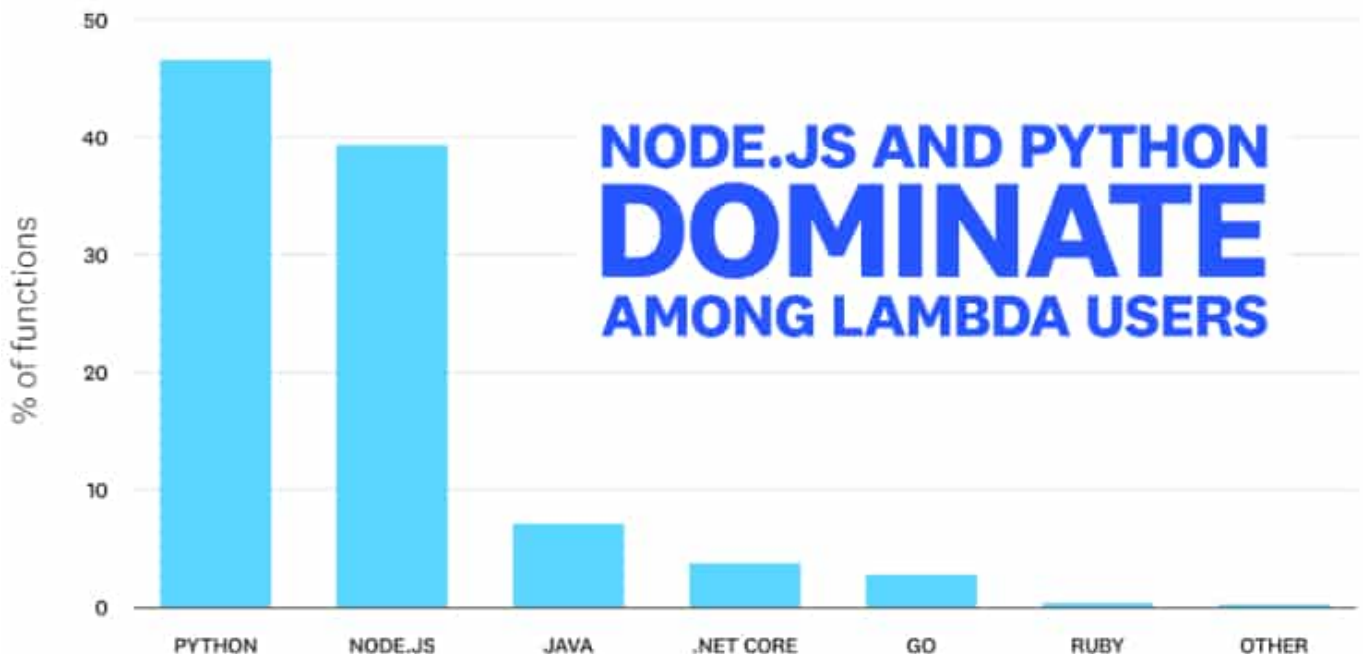
SQS and DynamoDB Are Popular in Serverless Architectures



Sur le volet des langages de programmation, 47 % des instances Lambda déployées tournent en Python ; 39 %, en Node.js.

Une distribution qui reflète l'évolution du service : Node.js fut le premier *runtime* pris en charge (en 2014), suivi de Java et Python (2015). C# (via .NET Core), Go et Ruby ont rejoint la liste en 2018.

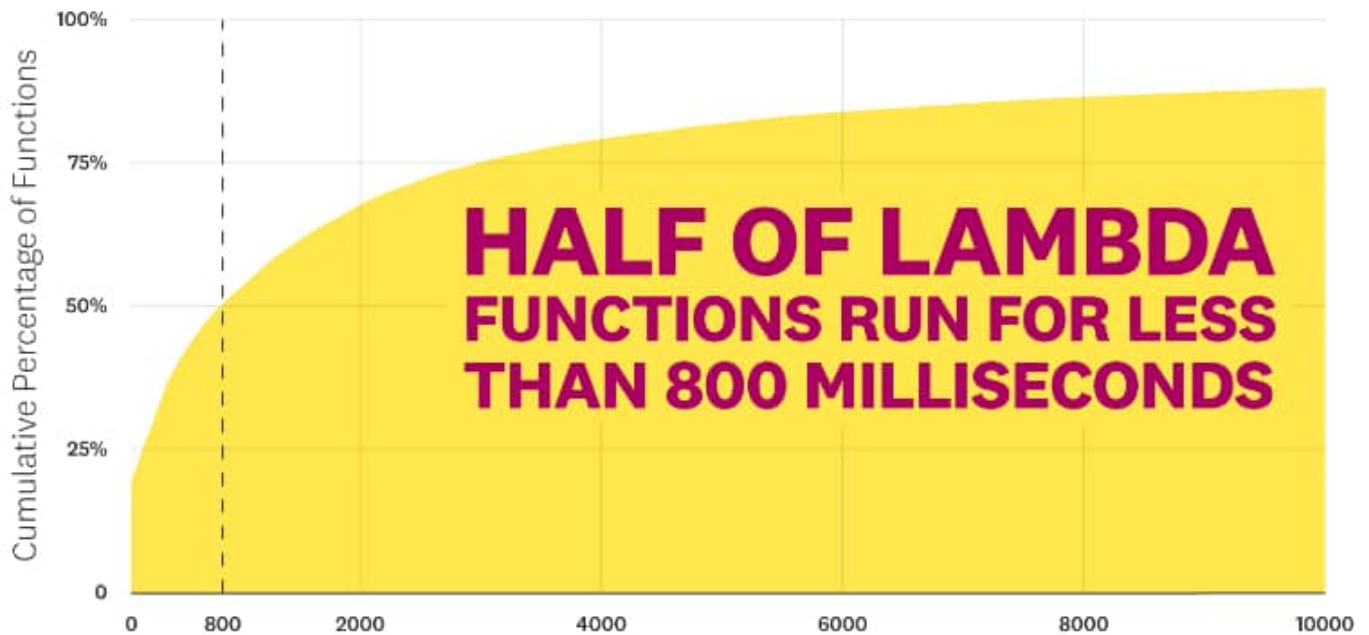
Most Popular Languages by Distinct Functions



Mémoire courte

La durée médiane d'exécution d'une fonction s'élève à 800 ms. Pour un quart, elle dépasse 3 secondes ; et pour 12 %, 10 secondes.

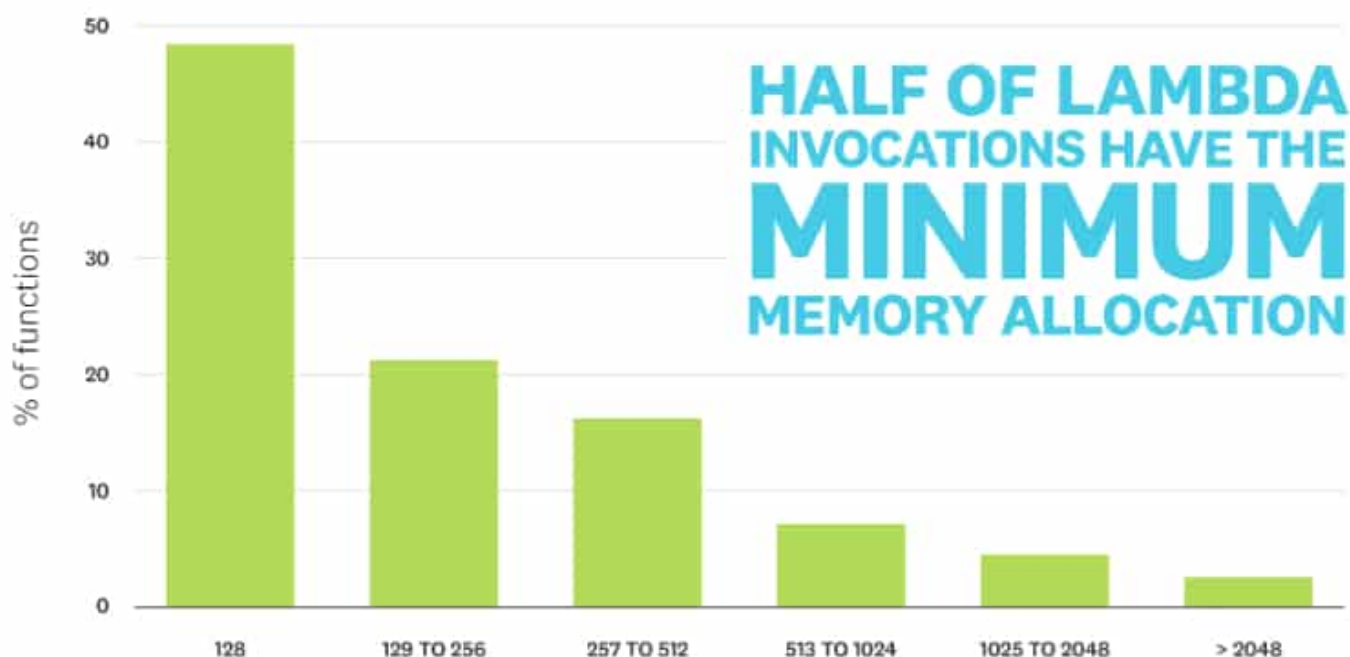
Duration of Lambda Functions



Le coût d'une fonction Lambda correspond au produit de la durée d'exécution et de la mémoire allouée.

Sur ce dernier paramètre, la tendance est à réduire au maximum les ressources : 47 % des fonctions sont configurées pour fonctionner avec 128 Mo de mémoire. Seules 14 % ont plus de 512 Mo à leur disposition (le maximum étant de 3 008 Mo).

Memory Allocated to Lambda Functions

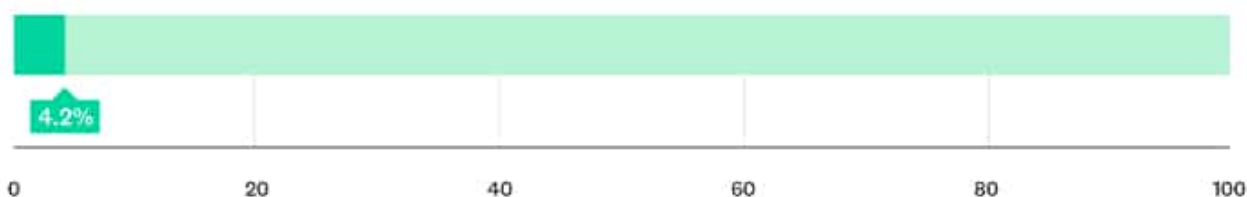


Par défaut, Lambda est limité à 1 000 exécutions simultanées de toutes les fonctions dans une région donnée.

Il est possible de définir une limite propre à une fonction, afin de lui réserver une partie de ces exécutions.

Près de 9 organisations sur 10 (88,6 %) ont cette pratique, mais à petit périmètre : au global, 4,2 % des fonctions ont une limite propre.

% of Functions That Have a Concurrency Limit Defined



VERY FEW LAMBDA FUNCTIONS HAVE A CONCURRENCY LIMIT

% of Orgs That Have a Concurrency Limit Defined for at Least 1 Function

