AWS DevOps Competency

Technology Partner Validation Checklist

October 2016
Version 1.0

Table of Contents

Introduction

The AWS DevOps Competency Technology Partner Validation Checklist is intended for APN Partners who are interested in applying for AWS DevOps Competency. This checklist provides the criteria necessary to achieve DevOps Competency for APN Technology Partners, under the AWS Competency Program.

The goal is to recognize APN Partners who help customers define and implement their business transformation journey leveraging the best in class DevOps consulting, architectural best practices, technology, and services required, based on the customers' business and technical needs.

APN Partners undergo a validation of their capabilities upon applying for AWS DevOps Competency, and every 12-24 months thereafter.

AWS reserves the right to make changes to this document at any time.

It is expected that APN Partners will review this document in detail *before* submitting an application for the AWS DevOps Competency, even if all of the pre-requisites are met. If items in this document are unclear and require further explanation, please contact your AWS Partner Development Representative (PDR) or Partner Development Manager (PDM) as the first step. Your PDR/PDM will contact the Program Office if further assistance is required.

Competency Application and Audit Process

In order to schedule the validation, APN Partners must submit the DevOps Competency application by following the steps outlined below:

- Step #1: Review AWS DevOps Competency: Technology Partner Validation Checklist
- Step #2: Submit AWS DevOps Competency Application through the APN Portal
  - Login to the APN Portal
  - Click "View My APN Account" in left navigation
  - Scroll to AWS Competencies and select DevOps Competency
  - Complete the DevOps Competency Application

Once your firm's application has been submitted through the APN Portal, the APN Team will review for compliance, then send the application to an AWS Partner Solutions Architect for technical assessment.

AWS recommends APN Partners have individuals who are able to speak in-depth to the requirements for the technical assessment.

Upon completion of the technical assessment, the AWS Partner Solutions Architect will submit a recommendation regarding Partner acceptance. The final decision regarding acceptance is made solely by the APN Team; APN Partners will be notified of their status by AWS.

Program Requirements

DevOps Competency Partners have demonstrated success helping customers evaluate and use the tools, techniques, and technologies of working with data productively, at any scale. Partners can obtain multiple DevOps competencies provided they meet the criteria for each category (listed below). For example, a Partner can achieve competency in DevOps – Infrastructure as Code and DevOps – Continuous Integration/Continuous Delivery if they have products that meet the requirements for each category.

AWS DevOps Competency – Technology Partner Requirements	
APN Membership	Advanced+ APN Technology Partner (view requirements)
AWS Support	Business level+ (view Support plans)
AWS Customer References	≥ 4 AWS Customer References specific to completed DevOps projects: • 2 of the 4 AWS Customer References must be public (i.e. case study, architecture documentation, whitepaper, blog post, etc.). ***Recommended***: Customer reference must demonstrate a large-scale deployment of the product typical to DevOps in terms of improving application delivery, application build/test, or infrastructure/configuration management.
DevOps Solution	**All Categories:** • Comprehensive documentation of the product or solutions capabilities, tools, and guidance for deployment • Product documentation includes details on integration with AWS • Product approved by AWS Architect Review Board (details submitted during application) • Product or solution meets AWS Security Best Practices • Public support statement on website that is easily discoverable • ***Recommended***: A whitepaper, website URL, or blog(s) highlighting Partner's solutions capabilities unique to AWS **Infrastructure as Code:** DevOps technology product or solution on AWS that offers components essential to infrastructure as code and configuration management: • Product or solution integrates with AWS services in a way that improves the AWS customer's ability to manage their infrastructure as code; including managing application configurations, infrastructure policies, containers, operating systems, and/or servers. • Integrates with one or more of the following services: AWS OpsWorks, AWS CloudFormation, AWS Config **Continuous Integration/Continuous Deployment:** DevOps technology product or solution on AWS that offers components essential to continuous integration & build or continuous delivery: • Product or solution integrates with AWS services in a way that improves the AWS customer's ability to build, test, or deploy AWS-based applications and/or

	• Integrates with one or more of the following services: AWS CodeDeploy, AWS CodeCommit, AWS CodePipeline, AWS Elastic Beanstalk, Amazon EC2 Container Service, AWS Lambda **Monitoring, Logging, Performance:** DevOps technology product or solution on AWS that ingests and/or analyzes logs and metrics, and/or monitors application and infrastructure operational performance: • Product or solution integrates with AWS services in a way that improves the AWS customer's ability to gain insight, trend and/or alert on operational or development metrics, logs, or application performance and/or • Integrates with one or more of the following services: Amazon CloudWatch, AWS CloudTrail, AWS CodeDeploy, AWS Config, AWS Elastic Beanstalk, Amazon EC2 Container Service, AWS OpsWorks, AWS Lambda, AWS API Gateway

AWS DevOps Competency Partner Validation Checklist

In preparation for the validation process, Partners should become familiar with the items outlined in this document, and prepare objective evidence, including but not limited to: prepared demonstration to show capabilities, process documentation, and/or actual customer examples.

1.0 AWS Customer References		Met	Not Met
1.1 Customer References	Partner has four (4) AWS customer references of the DevOps Two (2) references must be public. References must include: • Name of Customer and AWS Account ID • State of their business prior to your engagement • Problem statement/definition • Implementation of the solution for the problem statement • AWS services used ***Recommended***: One (1) of the customer references from 1.1.1 should provide specifics of the Implementation on AWS, in regards to one or more of the following: • Improving the application delivery lifecycle • Enhancing customer agility in infrastructure provisioning • Security measures being applied • Application/Infrastructure logs, metrics, or application performance information made available		

2.0 Solution Details		Met	Not Met
2.1 Solution Capabilities for DevOps	Partner must be able to demonstrate that their solution works with AWS for DevOps in terms of improving application delivery, application build/test, or infrastructure/configuration management. Partner, in addition, should ensure they meet the requirements for the specific category they are applying to by referencing the above competency requirements. They should also be able to demonstrate that the solution can scale to meet any size of the workload. Evidence must be in the form of either of the following components: • Product documentation of solutions available to public on how it helps customer with improving application delivery, application build/test, or infrastructure/configuration management. • For solutions offered as a product to deploy - Product documentation specific to AWS with clear guidelines on how to implement/use the solution on AWS. • For solutions offered as a service to use - Architectural details of the SaaS solution implemented on AWS. Details of the performance and availability to help customers optimize DevOps on AWS.		
2.2 AWS Best Practices	The product provides customers with DevOps solutions that are aligned with AWS architecture best practices and reference architectures. Partner must be able to provide an architectural overview that provides the following details on the use of AWS services like Amazon Virtual Private Cloud (VPC) and patterns like Multi-AZ deployments to provide highly available and reliable infrastructure.		

3.0 Security		Met	Not Met
3.1 Security Best Practices	Partner has documentation that proves that the solution is aligned as per AWS Security Best Practices and AWS Risk and Compliance.		
3.2 Security Control	Partner product or solution has the appropriate security controls: • For SaaS solutions, the solution does not in any form or control ask customer to provide their AWS credentials. All access controls are implemented via Cross Account roles. • For product being deployed by the customer, product has features to manage access to AWS resources and APIs using identity federation, IAM users, and IAM roles.		
3.3 Governance and Compliance	Solution has capabilities that support good governance and security, specifically including services such as: • AWS Identity and Access Management (IAM) • AWS CloudTrail		

4.0 AWS Services		Met	Not Met	N/A
In addition to their generic capabilities, Partner will undergo validation of their integration/usage of each services outlined below. For every service, there are a few guidelines/best practices outlined to ensure that the customer has a good experience with these services via the solution.				
4.1 AWS CloudFormation	• Credentials or secrets are not embedded in templates • Data should not be destroyed when a stack is deleted • Templates should not hardcode region-specific resources but instead rely on Mappings			
4.2 AWS Config	• Services consuming AWS Config records should not remove any of their information			
4.3 AWS OpsWorks	• Cookbooks used should be publically available and documented • Cookbooks should not contain credentials or secrets			
4.4 AWS Elastic Beanstalk	• Customer has ability to reference ebextensions in their application code			
4.5 AWS CodeCommit	• Credentials are not embedded for access to repositories • Accounts that have git command level access should not have control plane level access to the service • The root AWS administrative account should not have SSH keys set for CodeCommit access			
4.6 CodeDeploy	• Customers maintain control of deployment configuration • Auto-Scaling actions are paused during deployment to ensure consistency of application version • Rollback is enabled			
4.7 AWS CodePipeline	• There should be at least one stage that includes a testing action or a Manual Approval between Source and production environment Deploy stages. • Manual Approval user permissions should specific to that purpose			
4.8 Amazon CloudWatch	• Alarms are delivered via SNS to a minimum of one topic target			
4.9 AWS Lambda	• Log output from CloudWatch Logs should be relevant and consumed like other application logs • Instantiate AWS/Database clients outside of the handler for reuse • Monitor for and handle for errors such as throttles • IAM Roles are appropriately scoped for functions limiting permissions and access to services and resources as required • Functions do not have credentials embedded			
4.10 AWS CloudTrail	• Utilize CloudTrail with Events to minimize delay in actions compared to scheduled describe calls			

	• Store CloudTrail logs in a secure S3 bucket, and store copies outside of the account to ensure log integrity			
4.11 Amazon API Gateway	• Implement caching to preserve backend resources • Utilize IAM credentials or custom authorizers for securing the API • Implement meaningful HTTP response codes			
4.12 Amazon EC2 Container Service	• Follow considerations in https://d0.awsstatic.com/whitepapers/docker-on-aws.pdf • Customers have the control to deploy in multiple Availability Zones			
4.13 Amazon EC2 Container Registry	• Log API activity with CloudTrail • Utilize IAM Roles for access to repositories • Do not build sensitive data into containers			