WHITE PAPER

Extend your Cloud Environment to On-Prem with DuploCloud

DuploCloud

Table of Contents

While it is not possible to compete with a public cloud in terms of feature set, elasticity, scale, managed services, geographic reach and bursty workloads, there are cases where it makes sense to run part of your workloads in an on-premises environment. AWS recognizes the potential benefits of a hybrid requirement as detailed on their edge offering using AWS technologies called AWS Outposts (https://aws.amazon.com/outposts/). Microsoft with Azure stack has similar offerings where they provide an edge deployment using hardware and software managed by them.

In the context of this Whitepaper, we will refer to AWS Outpost synonymous to Azure stack

One of the key advantages of AWS Outposts vs the traditional on-prem IT deployments is the availability of application-centric AWS platform services like container management, big data analytics, single management interface for application deployment, monitoring and logging. See https://aws.amazon.com/outposts/features/ for an example of an access policy model to enable use of the rest of the platform services from cloud like S3, DynamoDB, SQS, Lambda, Azure Storage, Active Directory and so on. A unified approach to automation and security of infrastructure in cloud and on-prem enables the elasticity of workloads.

As enterprises continue their journey to cloud, they also learn some lessons in terms of variable costs that come with the pay-as-you-go model. For example, in the public cloud, you pay for compute, storage, data transfer, API calls, number of requests, IOPS, etc. It gets hard to predict the cost and eliminate some of these variable expenses that are workload dependent. In addition, some workloads are inherently better suited to running on-prem using an Outpost-like deployment.

Unfortunately, many organizations haven't been able to realize these benefits due to either their existing infrastructure or as a result of the cost barrier in terms of the minimum hardware spend and logistics of procuring and installing the hardware.

In this whitepaper we detail the subset of use cases and outline a software only implementation to get an AWS Outpost-like environment using standard commodity servers.

Software Only Outpost Deployment

One can deploy the smallest AWS Outposts for $6,965 per month catered to test and dev workloads. See details for larger, more feature rich and expensive stacks here: https://aws.amazon.com/outposts/pricing/

As an example, a 2 node (g4dn.12xlarge, 48 vcpus, 192GB RAM, 4 GPUs) GPU ready configuration is $8,914 per month. This comes to $282,000 if you pay upfront for 3 years. Similar hardware if procured with 4 GPUs, same number of cores, memory and storage will cost around $70,000 as one time expense for 3–5 years. On top of that you have to add up some of the cost in terms of datacenter, networking and management, to get the actual cost savings.

So if you want to take advantage of your own existing hardware or buy something specific to your application needs, you can't do that with Outpost. You have to use both hardware and software from the cloud vendor and you have no control over it.

Requirements for a software only AWS Outpost

We believe there are two requirements that, if met, can allow you to leverage on-prem servers even in a small quantity as an extension to your public cloud.

The two prerequisites for running an on-prem environment are as follows:

1. *Container Adoption or ETL jobs:* you should be able to either package and run your workload using a set of containers. Alternatively the workload can be a set of jobs in an ETL pipeline that the user wants to run against a big data cluster like spark.

2. *No low latency access needed to public cloud services:* If the application is cloud-native and wants to use platform services from the public cloud like DNS, load balancers, S3, DynamoDB, SQS, etc., while running the expensive compute on-prem, then the application should be tolerant to the latency incurred while the on-prem compute communicates with cloud services.

Why these two requirements, you may ask?

Simply put, the first requirement removes the need to have an expensive virtualized infrastructure on-prem and dealing with all the operational complexity that comes with it. The second requirement ensures that your application or parts of it can leverage the on-prem compute thereby offsetting the bulk of the public cloud costs. Here you can continue to use some of the non-compute cloud services like S3, SNS, SQS, etc., as long as accessing them over WAN is not an issue.

Once these two requirements are met, a few challenges still remain both in terms of dealing with the hardware and operational burden that comes with managing and running workloads on these servers.

Ideally you need management software to make the on-prem servers almost look and feel like cloud instances (like EC2 instances in case of AWS). If you can truly achieve that operational simplification, then the edge can look just like an extension of public cloud, while providing cost savings, higher performance and the locality that your team will love.

Use Cases For Edge Compute

We have provided an edge deployment successfully to several customers who were able to extend their AWS cloud to an on-prem infrastructure and cut their costs by 60% while getting higher performance as part of running native containers with local IO access.

Based on these existing customer scenarios, we have come up with 4 use cases where such an environment can help as long as you can minimize the operational cost.

1. Test and Dev Workloads
2. Big data analysis
3. AI/ML on local data
4. Standard IaaS based web hosting

Let's look at each in more detail:

1. *Test and Dev Workloads:* The test and dev workloads need to run 24×7 as part of a company's CI environment. They are not mission critical, they do not require reliable replicated storage and can provide fast local access to the developers. Builds can also complete faster due to lack of virtualization overhead and local disk access.

Running them on a set of servers on-prem can have several such benefits.

2. *Big data analysis:* Most big data analysis software relies on compute-intensive distributed processing like Spark, storage of large data sets, and finally light weight post-processing to produce the final results. These are mostly batch jobs and experiments that are not real-time.

   One could build an ETL pipeline that consists of compute intensive Spark jobs on-prem, then transfer the results to S3 followed by light weight post-processing via Lambda, and finally exposing the data in S3 via SQL AWS Athena. Again, you don't need reliable storage as data is typically replicated by the underlying NoSQL storage system like HDFS. Since you always pay for the data at rest in the cloud and pay extra for the IOPS, this can be a cost effective solution.

3. *AI/ML on local data:* Many companies are collecting terabytes of data per day from their AI/ML based applications. Examples are self driving car companies generating training data from their cars everyday, insurance companies going through a large number of documents, ride sharing platforms like Uber and Lyft going through millions of driver documents for verification everyday, and real estate or financial companies going through a lot of legal documents for verification. Much of this data is produced on the edge and needs to be analyzed there. You can continue to run some bursty training jobs either on-prem or in the public cloud. Ideally, there is a single cluster across local GPU servers and cloud GPU instances. You can then choose to run the AI workload on a specific location based on data and compute availability.

4. *Standard IaaS based web hosting:* This workload again needs to run 24×7 and does not require many public cloud features other than a simple load balancer, basic application monitoring and a periodic database backup. Again, these workloads can be easily migrated to a couple of edge servers where they can run in a much more cost-effective manner. You can continue to use ELB, Route53, WAF, CDN and monitoring features from the public cloud itself.

Table 1 below provides a high level summary of these workloads and the features that make them unique to benefit from an edge deployment.

Workload Type	Features that Make it Suitable for Edge
Test and Dev	Runs 24x7, no reliable storage needed, low-latency for dev and QA teams, native containers help with faster builds and testing.
Big data analytics	Works well with a bunch of servers with local disks, no need to pay storage cost when not running, faster with local network and bare-metal containers.
AI/ML	Run close to the data, multiplex hardware across teams, lower cost compared to cloud GPU instances
Standard IaaS based web hosting	Runs 24x7, burst can be absorbed by the public cloud, continue to use ELB, Route53 and WAF in AWS.

Table 1: Workloads and their characteristics that make them suitable for edge computing.

Challenges In Edge Deployments

Let's say you have some servers in a colo or in a lab inside your company. Once you rack and stack these servers, you need to go through a several of steps to make them consumable:

OS and Hardware Management

- OS installation: PXE boot, USB install
- Firmware install or upgrades
- Package install: using internal or external package repos

Application deployment

- Access to container registry: internal or external
- Application deployment: container management, spark and hadoop clusters

Logging and Monitoring

- Logging: Filebeat, Elasticsearch, Kibana
- Monitoring: Nagios, Prometheus, Grafana, InfluxDB

Security and Access Control

- Multi-tenant access to developers or teams from different projects
- Connecting on-prem deployment securely with AWS services
- Use role-based access control without leaking AWS keys

This all needs to be handled by some software layer or you have a big operational task on your hands to automate and make this environment really useful. This is exactly what AWS Outposts solves by providing a combined hardware and software solution managed by AWS. While this is still useful, having a software only solution that can work with any hardware is even more powerful and will offer a lot more flexibility.

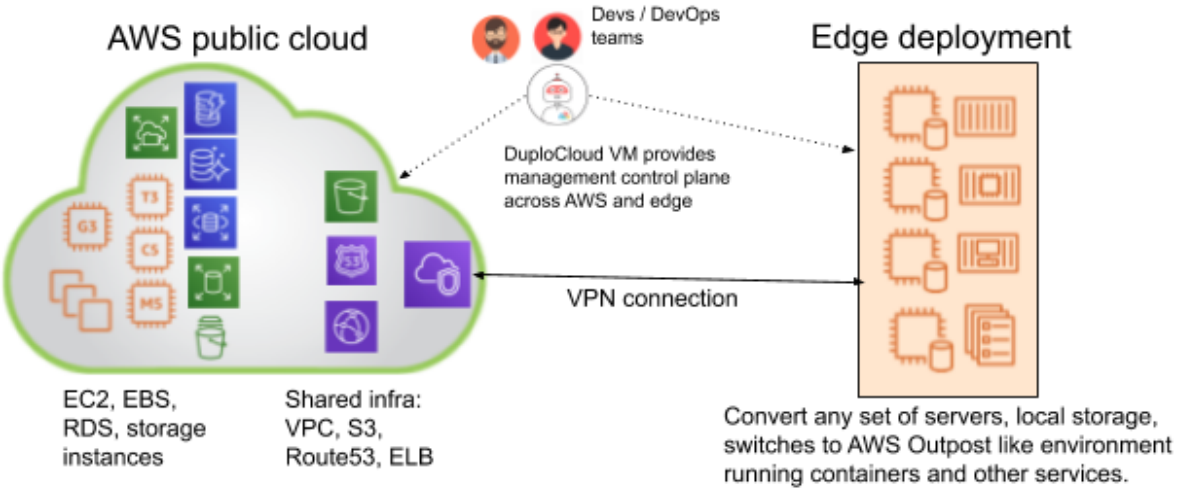
Imagine if you could simply install an OS on these machines and within few minutes, they could just be part of your AWS environment as a cluster, where you can deploy your application using containers, get monitoring, logging, multi-tenancy and upgrades done automatically by a management software running in the cloud itself as part of your own account?

The goal is to make these servers on the edge appear like EC2 instances with some tags and similar networking and roles to access other AWS services. This is where DuploCloud comes in and stitches all this together to create a seamless edge cloud. Let's take a look at this really impressive solution.

Seamless Edge Management with DuploCloud

With DuploCloud, we have built an edge computing solution that can convert any set of edge servers into AWS EC2 instances with local storage used as EBS volumes. You can simply install the operating system on the edge servers, which can be standard Linux or Windows installations and add them to the cluster managed by DuploCloud. Other machines in the cluster can be standard EC2 instances also. DuploCloud itself is self-hosted software that you can run as part of your own cloud account in AWS, Azure, or GCP.

Once DuploCloud takes over the machine, it will automatically install all the containers needed for logging, monitoring, security and compliance. It will also assign roles to the machine so that it can look and behave like an EC2 instance with specific access to the AWS resources. Now as you launch containers, which can be part of your CI/CD, big data analytics, AI/ML workloads or any general hosting, you can launch them either in the public cloud or an on-prem machine from the same management plane built by DuploCloud.



Operational Model

For a company that already has on-prem hardware, they can continue to use their racks, servers and teams to do minimal setup. For others, they can use a colocation facility that provides bare-metal servers and takes care of power, cooling, OS install & networking. Once the servers are available, DuploCloud will take care of installing all the packages related to monitoring, logging, security and compliance and show these machines as part of a built-in SIEM dashboard for your specific compliance requirements. The dashboard will also show you when some packages are out of date and need updating. The tasks that a customer has to do are:

- Install the operating system
- Upgrade any firmware if needed
- Upgrade any software packages on the host

These are all infrequent or a one-time operation which can be handled by anyone with basic understanding of Linux or Windows.

Everything else is taken care of by DuploCloud's management and orchestration software.

Conclusion

If you have any on-prem hardware that you want to use as an extension to your cloud environment, DuploCloud can help. You will also get built-in application deployment, CI/CD

flow, monitoring, logging, security controls, connectivity to your VPC and a SIEM dashboard for compliance. All these can take months to set up if done manually or even using automation scripts like Ansible, Chef or Puppet. If you choose to do automation yourself, getting a true extension of your public cloud account where local machines have secure access to cloud resources and services extra work that you will need to complete in order to have seamless deployments and the ability to burst.

If you want a seamless edge cloud but don't want to deal with all the operational challenges that come with it, please check out
https://www.duplocloud.com/out-of-box-edge-computing.html