Automating Cloud Migration with a No-Code & Low-Code DevOps Platform

DuploCloud

© 2021 DuploCloud, Inc.

Table of Contents

Introduction

As more and more businesses surge into the cloud as the foundation for digital transformation, they are looking for ways to streamline their processes, alleviate infrastructure complexities, avoid information leakage, and reduce costs. There are the new generation enterprises who were born in the cloud (Airbnb, Netflix, Snowflake), and then there are enterprises who have their workloads on premises in company-owned data centers. The term cloud migration refers to moving the on-prem workloads to public clouds like AWS, Azure and GCP. Cloud migrations can apply to two categories of applications:

- Custom Applications developed by Engineering teams
- Commercial off-the-shelf (COTS) Applications

Commercial-off-the-shelf (COTS) software is a term for software products that are ready-made and available for purchase in the commercial market. Examples of these are Microsoft Exchange Servers, SAP, SiteCore, etc. Many of these are now available as SaaS offerings by their respective vendors. For instance, Exchange is available as Office365. This whitepaper focuses on the migration of custom applications to the cloud, highlighting the work streams, different technical choices, execution phases, and the advantages of using DuploCloud throughout the process.

© 2021 DuploCloud, Inc.

Migration Work Streams

Migrating an environment from on-prem to cloud can be split into 4 workstreams:

1. **Application Migration.** This includes the custom-developed components by engineering teams.
2. **Data Migration.** This includes SQL databases, Stores procedures, jobs like SSIS, SSRS, file stores, and NoSQL.
3. **Infrastructure Migration.** This includes networking topologies, security groups, ACLs, IAM policies, user management, encryption keys, disks, SQL Servers, file store setups and the full list of compliance controls.
4. **Security Controls.** For businesses in regulated industries, there is a list of exhaustive controls to be implemented. Unlike other workstreams, the overall approach to security is very different in cloud compared to on-prem. T

The expertise required to replicate and migrate the infrastructure components is more-or-less standardized but requires cloud SME and can be very labor-intensive with current automation techniques. Application Migration is less laborious but involves subject matter expertise custom to the specific application. Data migration is partly standardized and partly requires understanding of the custom logic in the environment. For example, familiarity of Databases, their inter-dependencies, stored procedures, SSIS, SSRS etc.

Application Migration

There are three common approaches to application migration:

- Lift-and-shift Migrations
- Redeploy with no application or configuration change
- Repackage (Containerize) and Deploy

In Lift-and-shift migrations, the entire operating system is replicated in the cloud. Except for hardware drivers, the complete configuration across storage, compute, network and application is replicated. Network IP addresses may change. There is specialized software like Cloud Endure and Azure Migrate that make this possible. In the redeployment approach, the virtual machine, with a matching version of operating system, is launched in the cloud and only the application is replicated from the source. Finally, in the repackaging approach, the application components are packaged as docker images that are launched as containers in the cloud. Typically, there is no application code changes required, but the deployment process that includes setting up of configuration files and parameters would change. Figure 1 shows the main pros and cons of each approach.

Approach	Advantage	Disadvantage
Lift-and-Shift	Minimal change and typically fastest way to get the first version of the migration up and running	• Deployment is not "Cloud Native" i.e., it does not integrate well with cloud best practices around application, VM configurations that include availability, diagnostics, and scale • Hardest to maintain
Redeploy with no application or configuration change	Using a cloud native VM setup i.e., deployment is using the latest configuration in terms OS versions and diagnostics configurations like agents	• The application itself may still not be cloud native • Middle ground in terms of maintenance but largely a manual process
Repackage (Containerize) and deploy	Closest to cloud native. Most automated, easy to maintain and scale	• Requires sound subject matter expertise in both containers and infrastructure • Legacy technologies and stateful applications may not lend themselves to containerization

FIGURE 1

Data Migration

Data migration typically involves 2 types of data stores: SQL and File store. The most important element of data migration is to reduce downtime. Applications may be a 24x7 SAAS service with large amounts of data. Stopping all incoming requests for several hours or days while the data is copied over is not an option. There are 2 strategies for this:

- **Differential Copy over internet**. Here data is copied from on-prem to cloud while the application is functioning. The process is repeated several times over till we reach a point where the delta is small enough and can be copied over in a very small time. That process/action will require downtime to ensure no new data is being added to on-prem data stores. We have seen these techniques work well for a few terabytes of data. To speed up the data transfer one could add dedicated Site-to-site VPN connectivity with dedicated bandwidth rather than copying over the internet.

- **Offline copy by shipping disks:** This technique will be required when the size of the data is so large that it is practically impossible to copy it over the internet. For such cases, cloud providers offer a hardware storage device that gets shipped to the customer’s datacenter where data is transferred via the internal network. Then the device is shipped back to the cloud provider where the data is copied into the customer's cloud account.

For differential copy over internet, SQL databases provide two techniques:

1. **Differential backups.** In this technique, one would create a full backup, copy it over to a file store in the cloud provider and keep adding small differential backups at periodic intervals to the store. The idea is to get to a point where differential backups are so small that they can be created and copied over in say an hour or even less. When that point is achieved, the application is stopped; a final differential backup is then copied to the cloud and the whole series of backups is restored in an isolated database in the cloud.
2. **Secondary DB.** An extra database server is stood up in the cloud provider and connected to the primary as a replica. The primary begins to mirror all transactions' runtime. Over time, the secondary catches up to the primary. Finally, the application would need to be stopped briefly for the databases to flip over. This approach guarantees minimal downtime.

Differential copy of raw files is a very simple process using tools like Rich copy which can compare differences between two folders and copy only what is needed. Such a tool could be run repeatedly to replicate data from on-prem to cloud. The first replication would take the longest. Again, the goal is to reach a point where the last replication is achieved after stopping the application and takes minimal time to copy over.

Infrastructure Migration

Outside of the application packages and data, everything else constitutes infrastructure. 100% of cloud provider (like AWS, Azure) configuration falls in this category. Start by drawing out a high-level application architecture. This would typically be done by an architect in the organization. An example of one such diagram for AWS shown in Figure 2 below.

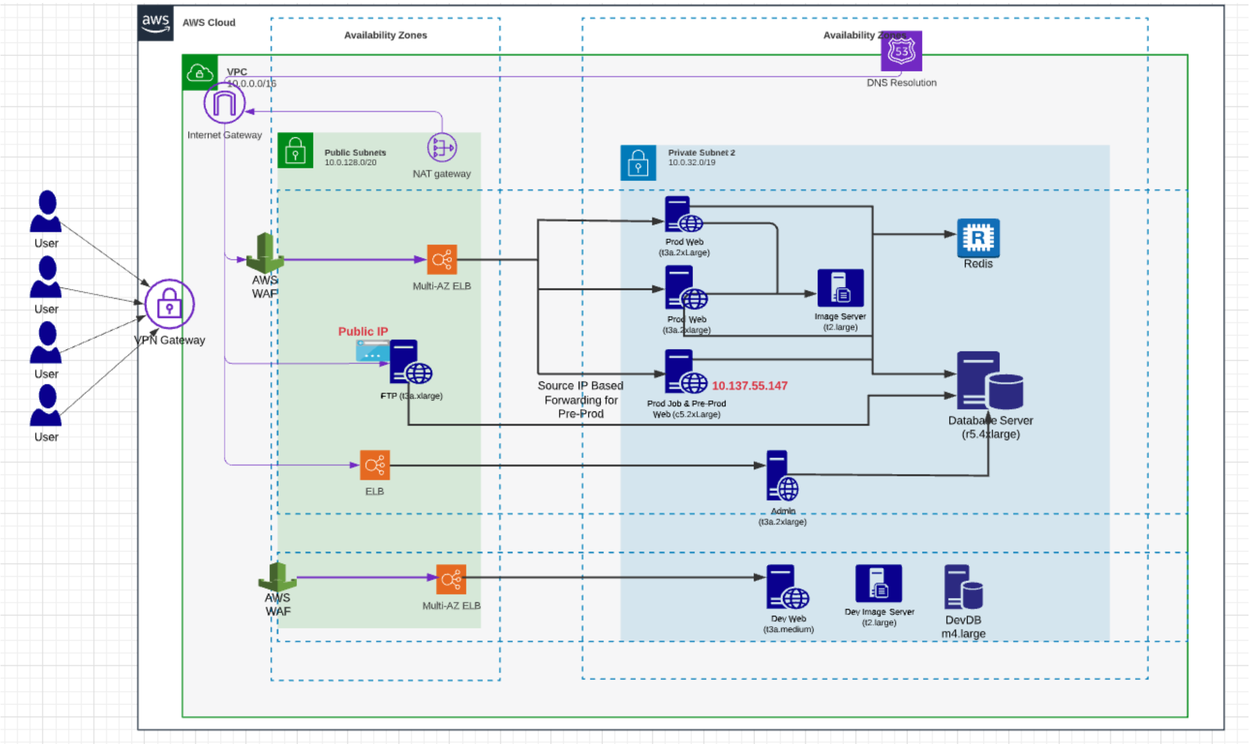

FIGURE 2

The topology consists of a VPC with a set of public and private subnets and multiple availability zones. The application is running in EC2 instances fronted by a load balancer and WAF with a MSSQL database and file store. There are multiple components of the application each with its own LB and WAF.

If an organization is in Azure, they have a deployment architecture that is Azure-specific. The constructs and terminology may change but conceptually it's the same. One such topology is shown in Figure 3.

© 2021 DuploCloud, Inc.

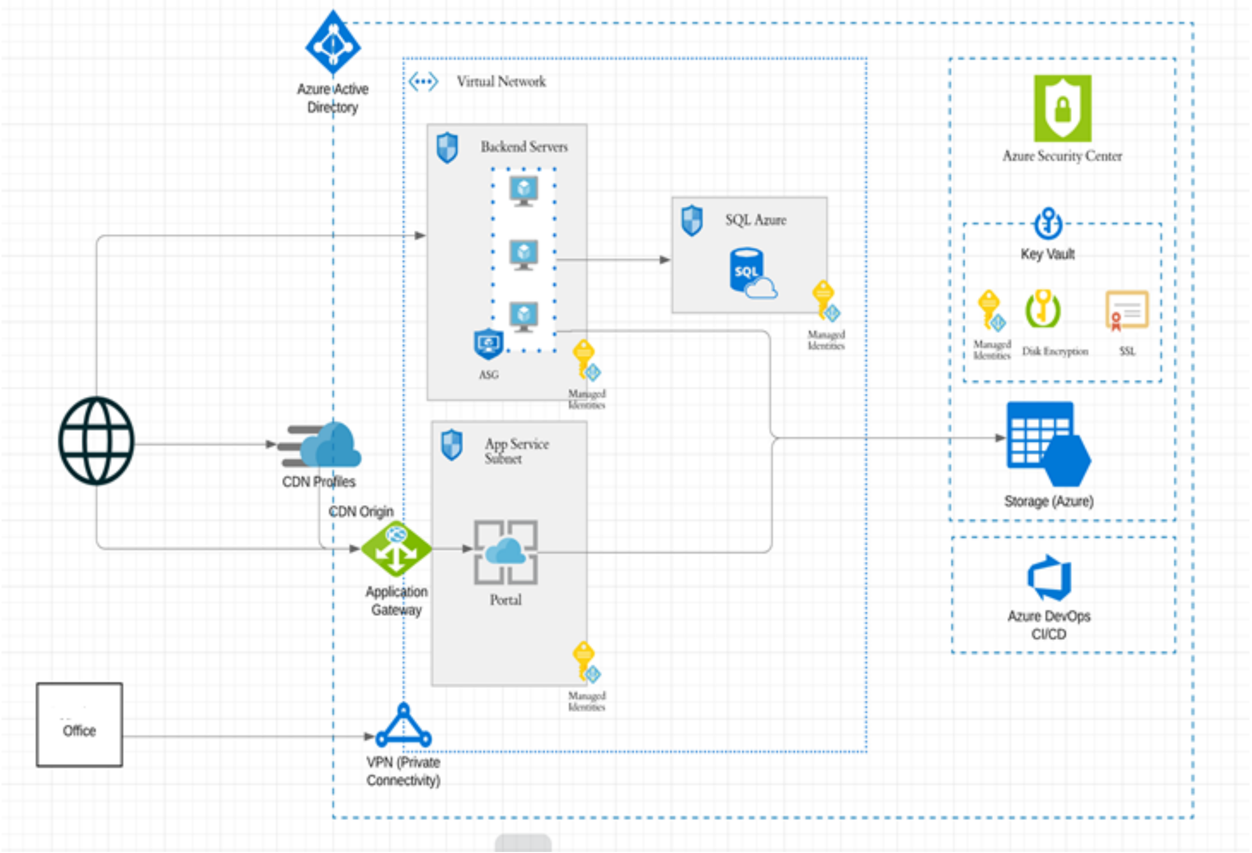

FIGURE 3

Such a high-level architecture, with some 15 odd constructs, gets passed to DevOps teams who translate these into hundreds of lower-level cloud configurations that would require thousands of lines of Infrastructure-as-code.

The infrastructure configuration to realize the application blueprints is typically done in a series of phases as shown in Figure 4.

Compliance Standard	Number of Controls
PCI DSS	79
HiTrust	80
HIPAA	50
AWS Well Architected framework	150

FIGURE 4

Security Controls and Compliance Standards

Intertwined with the infrastructure setup are the security controls. These can be the most labor-intensive component and thus deserves to be a workstream by itself. The process starts with the mapping of the desired compliance standards like PCI-DSS, HiTrust, HIPAA, GDPR, etc. with the corresponding configurations in the cloud. Cloud providers like AWS have published the mapping of these control sets which act as the authoritative implementation guide. In addition, there are tools like AWS security Hub and Azure Security Center, which hundreds of granular configurations that need to be applied.

Each compliance standard prescribes a list of controls as shown in Figure 4.

Using current automation techniques, each control would take a day or two to automate. One can see how security by itself can make migrations a multi-month project.

Following would be some examples of these controls:

- Turn off public access on S3 bucket
- Drop invalid HTTP headers in ELB
- Deploy firewall at each Internet connection and between any demilitarized zone (DMZ) and the Internal network zone
- Limit inbound Internet traffic to IP addresses within the DMZ.
- Rotation of secrets
- Isolation

An additional challenge with security is that unlike other work streams, the approach in cloud is substantially different. On-prem security is largely a centralized function revolving around firewalls, IDS, IPS, endpoint security and OS.

In cloud security, controls are distributed across the full stack of infrastructure. In addition, many expensive security tools being used on premise have limited applicability. Adapting to the new way of security is both a people and a process problem.

Automation

Automation of these workstreams is the single most important factor for any migration project. Duration of the project, downtime, security, and correctness are all functions of the level of the automation. Automation is achieved in a series of phases shown in Figure 5.

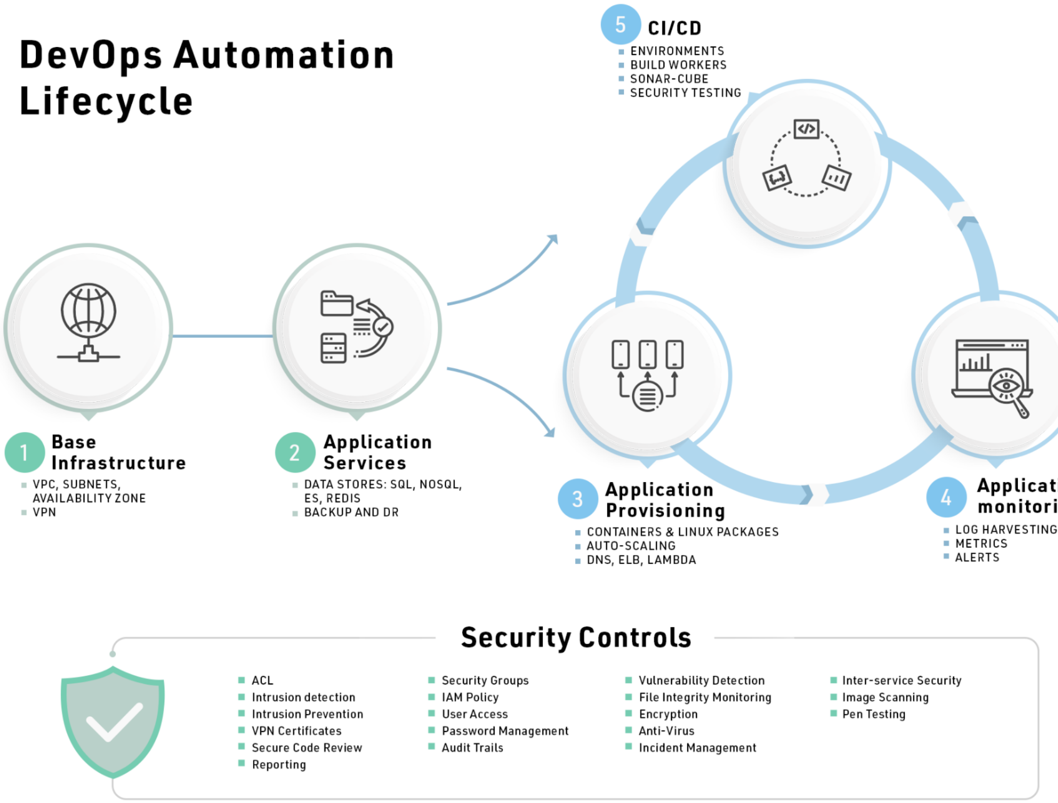

FIGURE 5

1. **Base Infrastructure.** This is the starting point where one would pick the regions, bring up VPC/VNETs with right address spaces, setup VPN connectivity and availability zones.
2. **Application Services.** This includes virtual machines, databases, NoSQL, object store, CDN, Elasticsearch, Redis, memcache, message queues, and other supporting services. Further in this area are disaster recovery, backup, Image templates, resource management, etc.
3. **Application Provisioning.** Depending on the application packaging type, different automation techniques and tools can be applied. For example, Ansible can

© 2021 DuploCloud, Inc.

automate deployments of vanilla virtual machines; Kubernetes for containers; and EMR and Databricks for data science workloads.

4. **Logging, Monitoring and Alerts**. These are the core diagnostic functions that need to be set up. Centralized logging can be achieved by Elasticsearch, Splunk and Sumo Logic, and Datadog. For monitoring and APM we have Datadog, CloudWatch, SignalFx, etc. and for alerts there is Sentry. Many unified tools like Datadog provide all 3 functions.
5. **CI/CD**. There are over 25 good CI/CD tools in the industry, from Jenkins to CircleCI, Harness.io, Azure DevOps, etc.
6. **Security Controls**. These are required across the board.

Executing the Migration

Once we have the migration strategy planned for the workstreams and automation in place, it is time to execute. Execution is performed in the following steps:

1. **Staging Environment**. A mirror of the running setup on premises is replicated. It starts out by deploying the underlying infrastructure, followed by creating the data stores which have a small but representative dataset and then finally the application is deployed in the environment. The environment is then validated. Security is largely a behind the scenes function and should ideally be baked into the automation.
2. **Data migration.** Start a data copy and reach a point where enough data has been copied over to the cloud so that any remaining data can be moved in a short duration by stopping the application to avoid any new data ingestion.
3. **Production Infrastructure Setup.** Bring up the production environment as per the blueprint pointing to the new data set. Keep the application in stopped mode.
4. **Begin Downtime and Finish Data Migration.** The on prem application would be stopped and the final data copy is performed.
5. **Bring up Cloud Environment.** The new production environment in the cloud is fired up and validations run. During this time, it is important to turn off any background jobs that can trigger production functionality. Once validation is complete the DNS names are swapped, any other finishing tasks performed and the environment is live in the cloud.

Automating Infrastructure Setup and Security

At DuploCloud, our focus is the Infrastructure Migration and Security Controls work streams. We have created an E2E automation platform that enables the DevSecOps lifecycle show in Figure 5 out-of-box. There are over 200+ tasks automated across the full lifecycle with a breakdown shown in Figure 6. The platform covers almost all configurations in a cloud provider and a multitude of compliance standards. Additionally, the power of the platform is not limited to the workflows and compliance controls available out-of-box. The innovative rules engine powering the platform enables newer services and workflows without having to write thousands of lines of code by expensive subject matter expertise.

Using DuploCloud for a cloud migration project accelerates infrastructure setup and security workstreams by a factor of 10, with a 75% reduction in costs. DuploCloud levels the playing field for cloud native and on-prem companies by making niche automation techniques like IAC easily accessible and adoptable by operations teams of all skill levels.

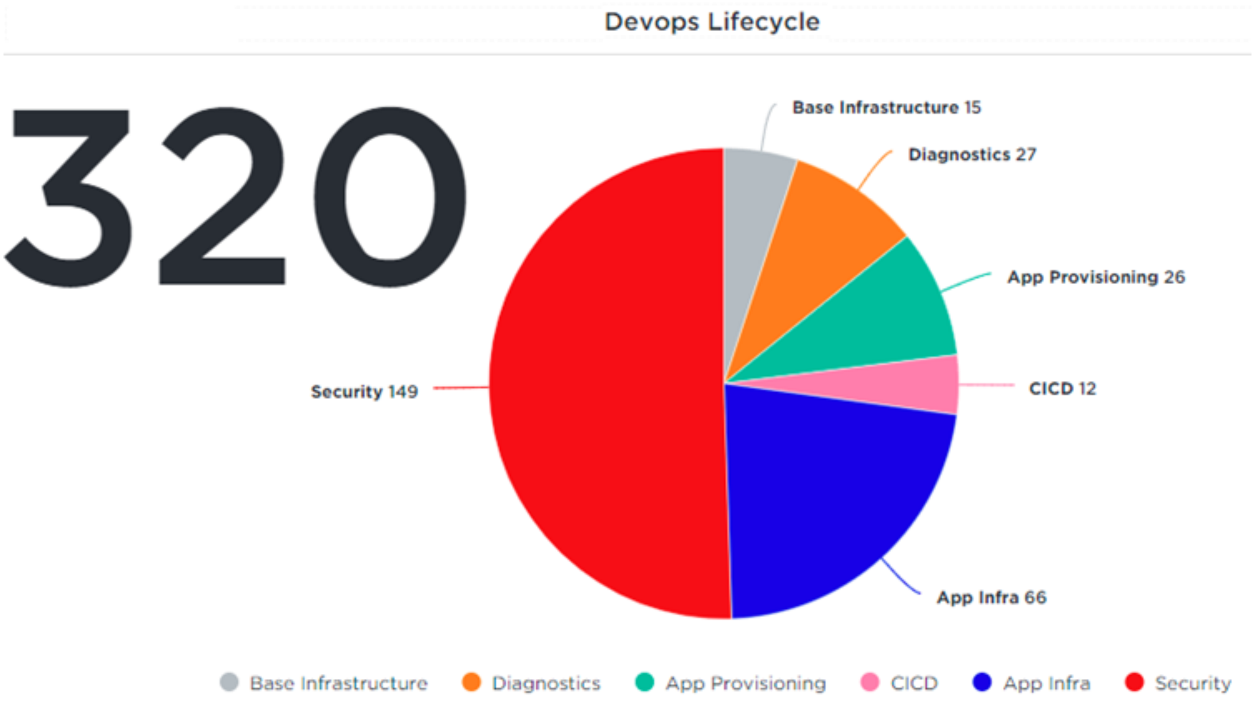

FIGURE 6

© 2021 DuploCloud, Inc.

Conclusion

Cloud migration requires careful planning, coordination, and automation. There are 4 key work streams: Application Migration, Data Migration, Infrastructure Setup, and Security Controls. The onus of application and data migration components is largely on the authors of these components. The infrastructure and security controls are more standardized.

While application and data migration require a custom skill set, it is a bounded component relative to infrastructure and security which are a lot more labor-intensive. Infrastructure Automation can substantially reduce migration times and improve operational efficiency post-migration. It is best done early in the process in order to have a scalable digital transformation path post-migration as well. At DuploCloud, our goal is to make automation and security a no-op so engineering teams can focus on the product rather than spending precious time and effort on standardized configurations.