Azure DevOps

Key Benefits Around People, Process and Automation

Introduction

Productivity of the software development teams has been one of the most critical challenges faced by enterprises in modern times. Microsoft leveraged 15 years of its learnings, tools, and investment into this product – Azure DevOps - and rolled out for businesses, big and small, to enable developers ship superior-quality software faster.

Launched in October 2018, Azure DevOps (formerly Visual Studio Team Services launched in 2006) is a SaaS platform that offers a full spectrum of DevOps toolchain for building and deploying software. It also integrates with many available tools on the market and has successfully addressed specific needs of many customers worldwide, irrespective of their language, platform, or cloud.
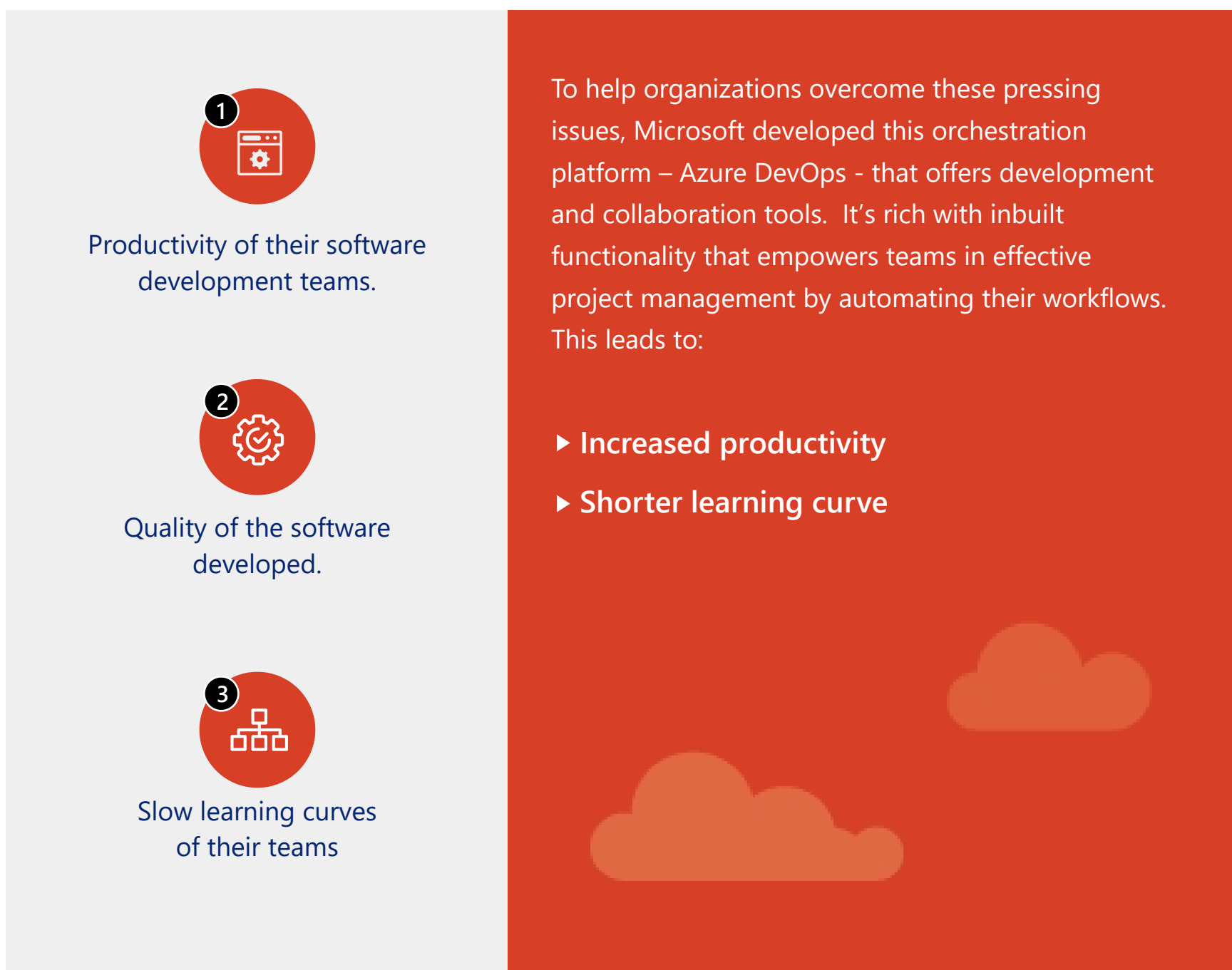
This playbook is meant for all the senior executives - in project management, research and development and application support roles - looking to tap into this massive opportunity and work in the DevOps environment on Azure. This playbook also provides some recommended best practices for those who are a little further along on their Azure DevOps journey.

Part I: What's Azure DevOps?

1. The Big Picture

Most enterprises, big or small, today grapple with three key challenges:

①

Productivity of their software development teams.

②

Quality of the software developed.

③

Slow learning curves of their teams

To help organizations overcome these pressing issues, Microsoft developed this orchestration platform – Azure DevOps - that offers development and collaboration tools. It's rich with inbuilt functionality that empowers teams in effective project management by automating their workflows. This leads to:

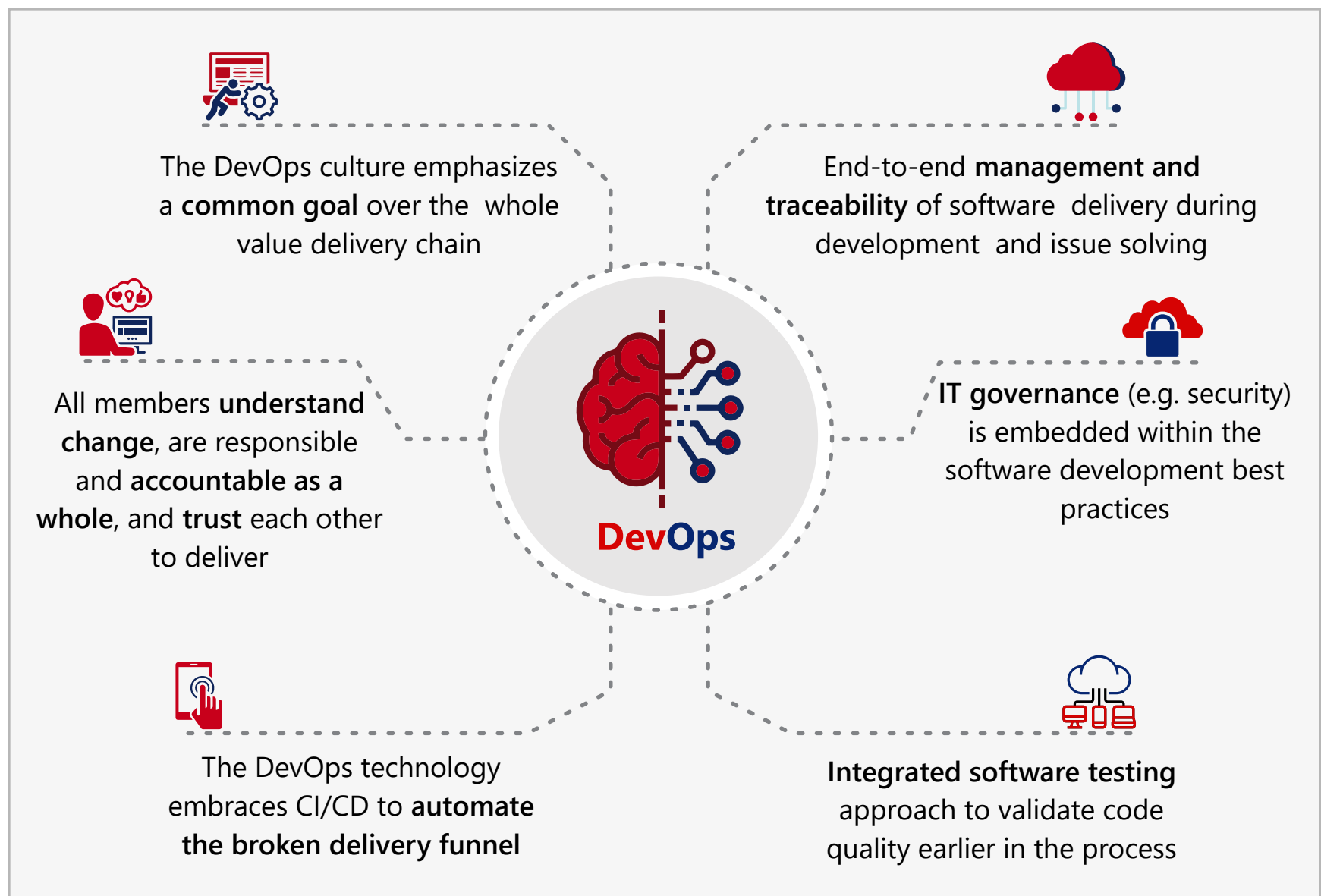▶ **Increased productivity**

▶ **Shorter learning curve**

a) A Snapshot of DevOps culture

A growth mindset is a must-have for DevOps teams. It all starts with the organizational culture and the people, who play a critical part in building it. Building a DevOps culture calls for significant Change Management around people, processes, and tools. But once the DevOps culture is created, it leads to a high-performing team environment.

DevOps unifies the mindset of Development and Operations

Today businesses prefer to release on-demand. With DevOps, both functions collaborate continuously to align business demands within the software delivery lifecycle.

The DevOps culture emphasizes a **common goal** over the whole value delivery chain

End-to-end **management and traceability** of software delivery during development and issue solving

All members **understand change**, are responsible and **accountable as a whole**, and **trust** each other to deliver

DevOps

IT governance (e.g. security) is embedded within the software development best practices

The DevOps technology embraces CI/CD to **automate the broken delivery funnel**

Integrated software testing approach to validate code quality earlier in the process

Applying agile practices, DevOps culture focuses on building small, multidisciplinary teams that can take ownership of projects and collective accountability for delivering the users experience of their software.

There are three key tenets of DevOps culture:

Collaboration, visibility, and alignment

Embracing a growth mindset

Allowing technology to drive change

There are three key tenets of DevOps culture

Key DevOps roles and responsibilities

Several key roles should be represented in a cross functional DevOps team; a team member with a T-shaped profile can fulfill more than one role

b) The DevOps Equation

Understanding the DevOps equation is critical to the success of DevOps culture within the organization. If you focus majorly on tools, it's likely to doom due to the DevOps Equation. It's not an off-the-shelf solution that can simply be implemented in a jiffy. The people side and their acceptance are critical yet apparently seem to be the least defined part of the equation.

 DevOps Equation = Quality x Acceptance

You could have a well-designed solution but if it is accepted by only a few of the stakeholders, then it won't yield much value to the organization. In fact, it's highly likely to fail.

The Azure DevOps Environment

Azure DevOps is a platform that enables the integration, testing, delivery, and deployment of applications and services to any cloud (using Azure DevOps Services) or on premises (using Azure DevOps Server). It integrates with other widely used services like Campfire, Slack, Trello, and more.

The Azure DevOps Environment helps businesses:

Improve dependency availability

You need to have the complete application and all its dependencies available for testing. Microsoft Azure partners like us can help you virtualize APIs and UIs for complete pipeline

Dramatically cut on the testing time

You can run your tests in parallel on the cloud with CrossBrowserTesting to reduce the build and test times.

Track your complete pipeline

Track the performance of your applications in the pipeline. It allows you to monitor the local projects too.

a) The Azure DevOps Process

b) Azure Deployment Cycle

Providing continuous value to customers, people, and processes, Azure deployment lifecycle can be divided into four distinct phases:

Plan

Azure DevOps leverages agile practices to empower teams to manage their projects efficiently and have complete visibility into products and projects. Chart out, monitor and prioritize tasks with Azure Boards.

Develop

Use Visual Studio and Visual Studio Code to code smarter and share with other like-minded developers. Create environment for developers in a few clicks

Deliver

To deliver more value to customers, you can deploy your application automatically on any Azure service in multiple cloud environments. You can create delivery pipelines in such an environment with tools like Jenkins.

Operate

Conduct full-stack tracking, receive actionable notifications, glean insights, and manage your cloud environment. Keep your applications compliant and check threat exposure.

STEP 1 — **Devise a Shared Tools Strategy**

A common tools strategy needs to be formulated to facilitate cross-team collaboration across development, testing, and deployment. It should reflect your business objectives for Azure DevOps and entail:

- Processes
- Development tools
- Communications & collaboration planning
- Deployment tools
- Operations & CloudOps tools
- Testing tools
- Integration tools

Developers must be able to send upgraded software to deployment and operations without any manual intervention.

STEP 2 — **Leverage tools for automating all requests**

Azure DevOps must capture all requests for changes in the software or even a new one. It automates these requests that come from the business or other development teams. No ad hoc requests outside the DevOps process are entertained.

STEP 3 — **Leverage agile Kanban project management framework for automation**

Azure DevOps must capture all requests for changes in the software or even a new one. It automates these requests that come from the business or other development teams. No ad hoc requests outside the DevOps process are entertained.

STEP 4 — **Understand the productivity metrics of both manual and automated processes**

Choose the tools that give you clear visibility into the productivity of your DevOps processes, both automated and manual. These tools will also help you understand if the processes are working in your favor.

Zero down on to the metrics pertinent to the DevOps processes. Also, define automated processes to rectify issues without manual intervention.

STEP 5 **Implement test automation and test data provisioning tooling**

Test continuously with Azure DevOps. Put the code into sandbox and run multiple tests. Once completed, it will automatically promote the code down the DevOps process, or send it back to the developers for working again.

STEP 6 **Run acceptance tests for all deployment tooling**

Clearly define the acceptance tests that will be a part of each deployment. Also mention levels of acceptance for the applications, data, and the test suite. Spend time defining the acceptance tests and ensuring that the tests meet with the acceptance criteria selected. Plug in the new requirements into the software as the applications evolve over time. The software must also be tested against these new requirements.

STEP 7 **Provider regular feedback in a cross-team setting to identify gaps, issues, and inefficiencies**

Use feedback loops to automate tests that identify gaps and issues. Flag the issue with the artifact to provide required information about its complete occurrence to the developers or operators. Also, establish the communication that takes into consideration all the human and automation tenets in the loop. This calls for cross-team collaboration to rectify the issue and a consensus for the type of resolution needed and the technology needed to rectify the issue. Then in the production phase, the tool must help you to devise the tracking process.
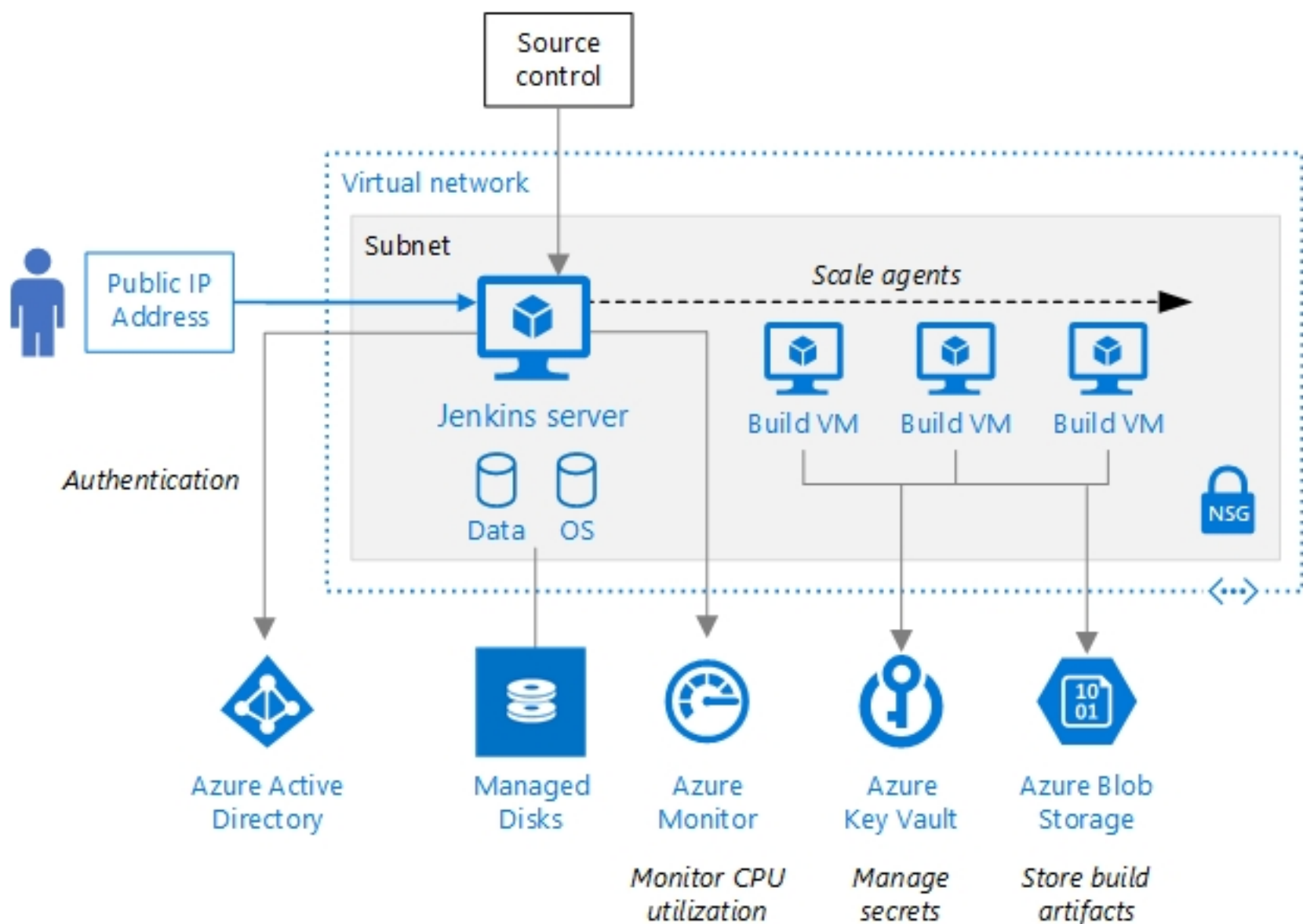
Part II: Automation Tools

Azure offers many ways of interacting and automating things. Various automation tools help businesses provision and manage Azure resources and embrace Microsoft Azure fully in an optimal way.

Let's look at some of the most widely used Azure automation tools below:

i. Jenkins on Azure:

Many developers have set up CI/CD pipelines on Jenkins for it is hosted internally by choosing from over 300 Jenkins Plugins available. These plugins help in creating and testing of their code as easily as possible. Good documentation is available. It enjoys massive popularity and bigger approvals (from 1529 developers), with mentions across 1775 company stacks.

Jenkins is an open-source automation server you can use to run continuous integration and delivery (CI/CD) for your applications. It's an excellent tool to build, deploy, or launch anything async. Classified as a "Continuous Integration" tool, Jenkins integrates with tools like Slack, Datadog, BrowserStack, Azure DevOps, Rancher, SonarQube, and many more.

Jenkins offers the following features:

Easy installation	Easy configuration	Change set support

Companies like Facebook, Netflix, Udemy, Instacart, Twitch and Lyft use Jenkins.

ii. Azure BluePrints

Azure Blueprints empowers IT professionals deploy a repeatable collection of Azure resources (templates and other Azure artifacts) defined by specific standards. It helps developers quickly create and set up new environments within the broader organizational compliance using built-in tools for accelerating the development and delivery processes.

Azure Blueprints currently supports the following artifacts:

Resource	Hierarchy options	Description
Resource Groups	Subscription	Create a new resource group for use by other artifacts within the blueprint. These placeholder resource groups enable you to organize resources exactly the way you want them structured and provides a scope limiter for included policy and role assignment artifacts and Azure Resource Manager templates.
Azure Resource Manager template	Subscription, Resource Group	Templates, including nested and linked templates, are used to compose complex environments. Example environments: a SharePoint farm, Azure Automation State Configuration, or a Log Analytics workspace.
Policy Assignment	Subscription, Resource Group	Allows assignment of a policy or initiative to the subscription the blueprint is assigned to. The policy or initiative must be within the scope of the blueprint definition location. If the policy or initiative has parameters, these parameters are assigned at creation of the blueprint or during blueprint assignment.
Role Assignment	Subscription, Resource Group	Add an existing user or group to a built-in role to make sure the right people always have the right access to your resources. Role assignments can be defined for the entire subscription or nested to a specific resource group included in the blueprint.

Azure Blueprints allows the IT professional to orchestrate the deployment of resource templates and other Azure artifacts, including role assignments, policy assignments, resource groups, and resource manager templates.

iii. Chef on Azure

Chef is a point-and-click configuration management solution you can create within the Azure portal. This point-and-click solution helps you to package and test your applications, upgrade your infrastructure, and keep it compliant and secure.
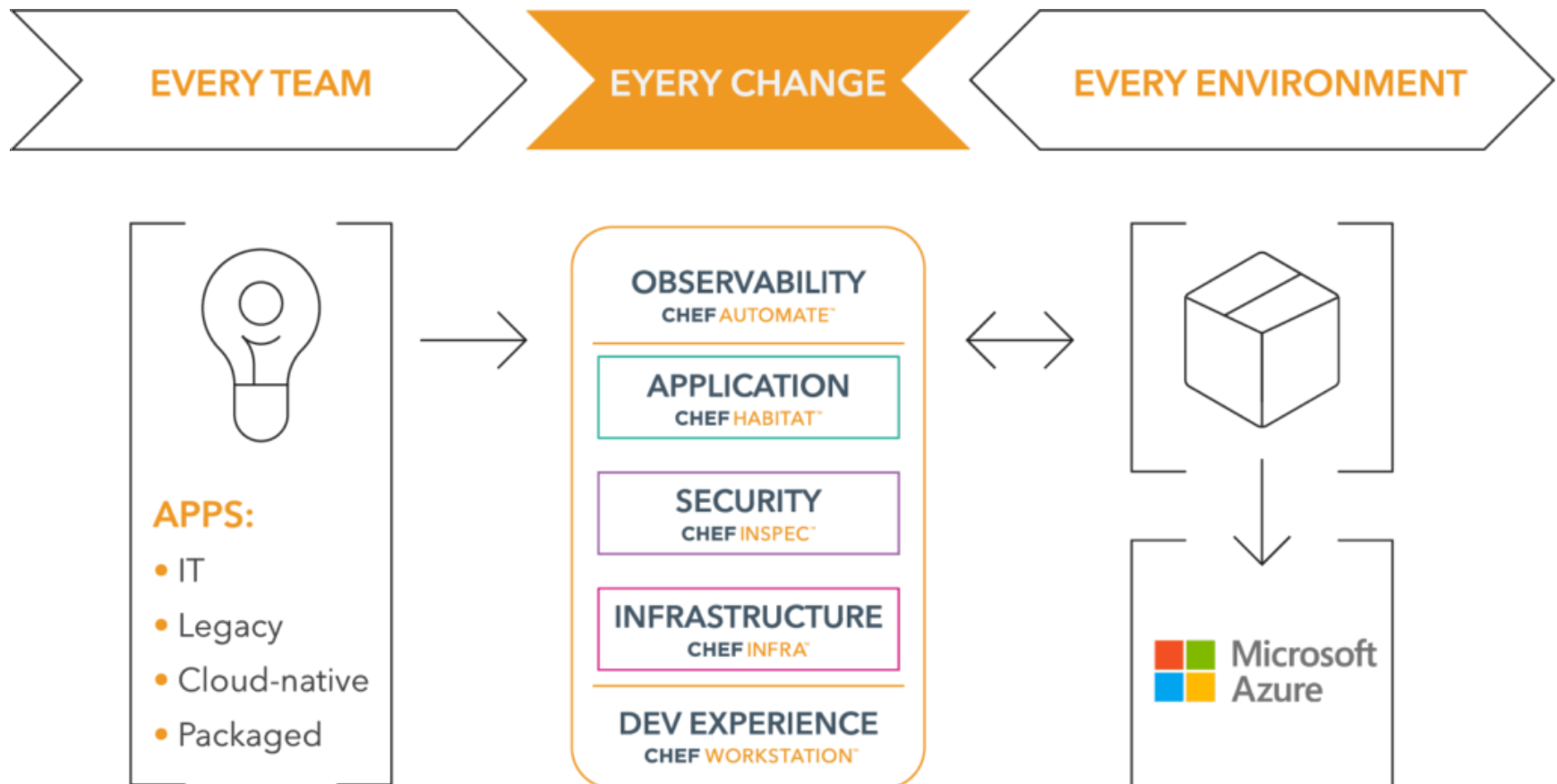
It automates virtual machine (VM) infrastructure, configures, deploys, and converts it into a code. This solution provides companies the tools they require to migrate smoothly to Microsoft Azure as these eliminate the hassles of migrating to the cloud. These also provide an in-depth understanding of the state of assets even before the migration occurs. This helps the customers migrate at the correct pace for their businesses.

Chef is planning to roll out an integration of its InSpec security and compliance tools with Azure. It will help the users to audit all the Azure applications effortlessly.

Ansible

Ansible is a configuration management tool that helps you automate provisioning (of VMs, containers and networks), configuration, and application deployments in your environment. It includes Ansible modules that run directly on remote hosts or via playbooks, and control resources like services, packages, or files.

It can configure both Linux and Windows operating systems and includes a declarative language to signify system configurations.

Ansible can be easily used to automate the same workloads in Microsoft Azure that supports hybrid cloud in the areas of infrastructure, user identity, and management. It, thus, gives businesses the flexibility to run workloads where they're most optimally used.

To use Ansible with Azure, you must clearly define the connection settings between the Ansible server and Azure. There are three pre-requisitely to get started with using Ansible, namely:
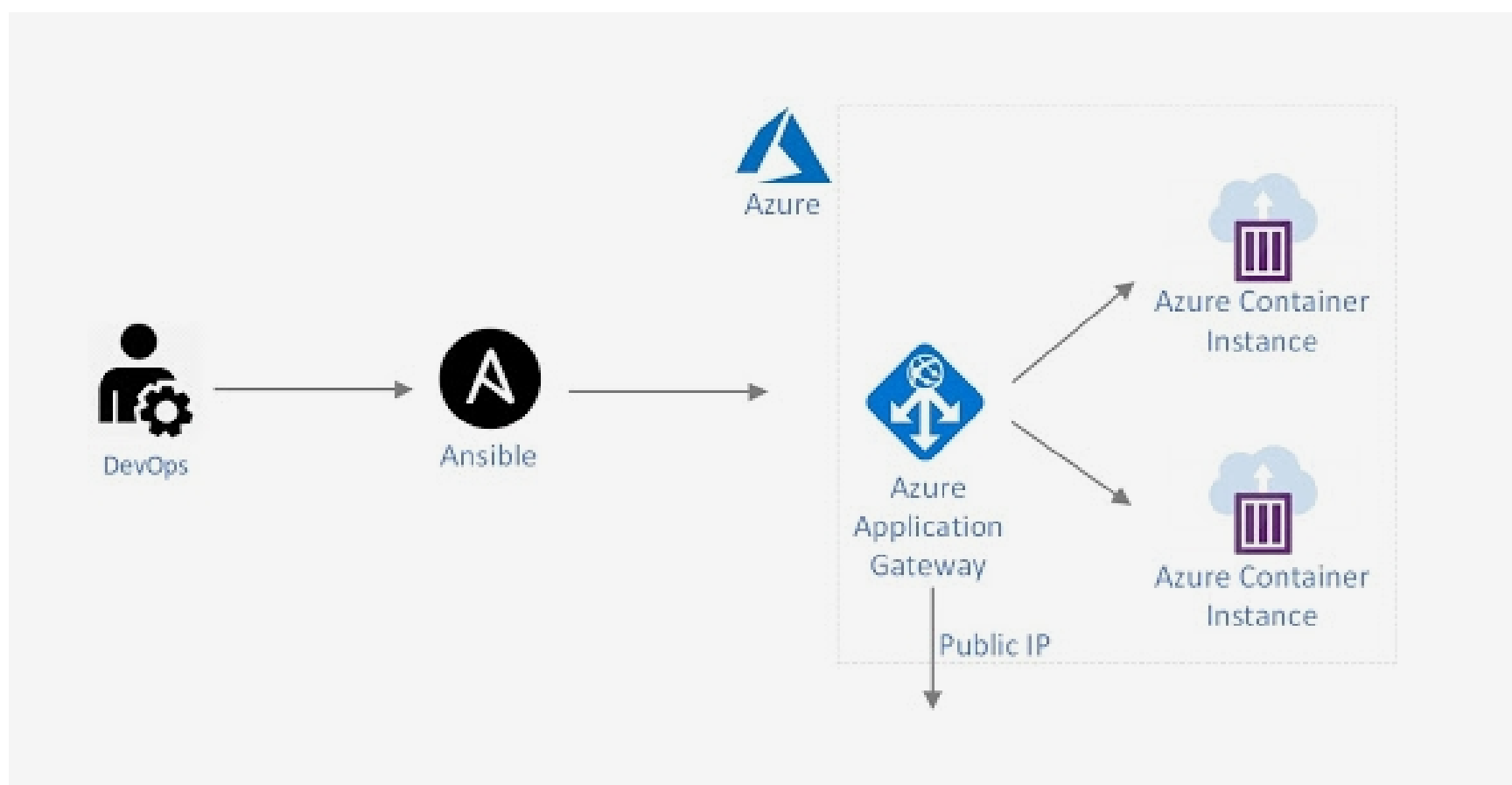
Azure subscription	Ansible installed for Azure	Azure PowerShell module

Conclusion

A robust DevOps practice and setup is vital for successful agile delivery and minimal risks. DevOps greatly optimizes release management costs and team productivity, enabling reduced time to market. Reach out to us to explore how we are helping our customers leverage Azure DevOps and different automation tools in their DevOps transformation journey.

Contact us:

Cambay Digital, 1838 Snake River Road, Suite A Katy TX 77449
Email: allen.scott@cambaydigital.com