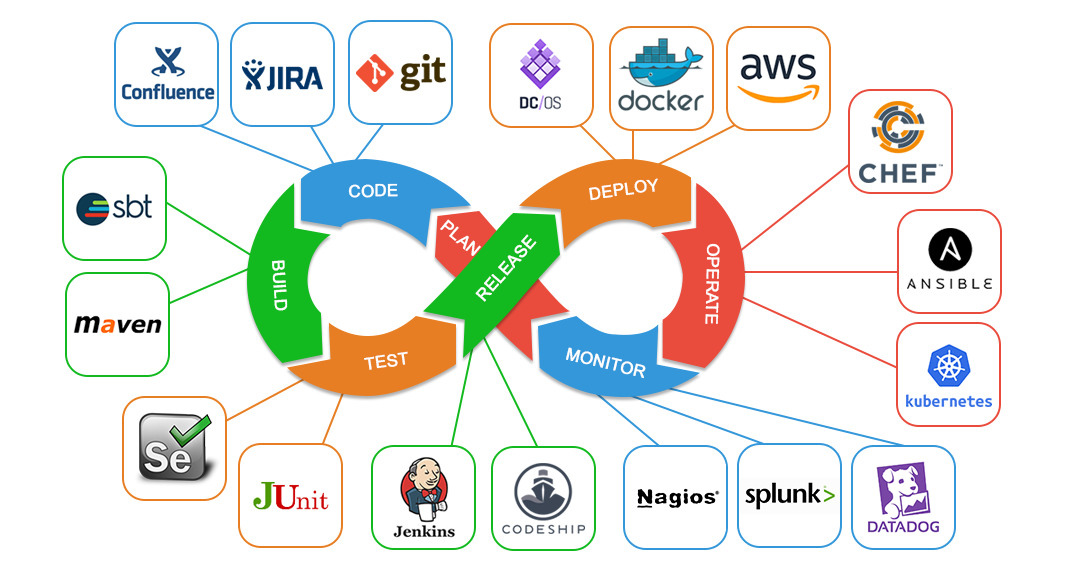
## Lecture 13 – Commonly Used DevOps Tech Exposed



We know of the potential pitfalls and perils when an engineer puts focus on tooling rather than the problems that need solving. We also know of the core areas that specific tech is used to provide an improved environment and workflow. Looking once more at the core areas before we dig deeper, we know that DevOps tools are used for:

* Project Management
* Version Control
* Containerization
* Cloud Infrastructure
* Configuration Management
* Container Orchestrion
* Metrics, Monitoring and Analytics
* Build Suites
* Test Suites

We’re going to be unpacking each one of the above categories and looking at the technology available for engineers to use in the field to bring optimization, progress, reliability and stability to the software development lifecycle (SDLC).

Project Management

When working amongst several different teams encompassing many individuals, it’s crucial for there to be some time of time and project tracking software. This allows project managers to have transparency insights as to what’s going on, for developers to know what tasks/features need developing and to provide a historical log of what’s been completed/released, amongst many other benefits which we’ll be examining. Lifting the veil on some of the project management products commonly found in the field, we’re going to explore why and how they’re used.

The Atlassian Suite

Knowing that DevOps is responsible for helping communication and collaboration to achieve optimal performance between development and operations teams, one of the fundamental useful pieces of software an engineer will be familiar with besides a VCS is a project management software suite. Atlassian products reign supreme in this domain, providing an entire suite much like Google’s SaaS business suite, Atlassian offers a plethora of products intended to help drive business operations. Their products not only foster collaboration, stimulate innovation, accelerate releases and continuously improve quality but they also are super extensible. Offering all types of web hooks, plugins and a feature rich product, Atlassian is usually what organizations will adopt to help bridge any communication gaps.

Offering a few different products, let’s see some of the most popular ones used and which ones you may commonly see once in the field.

Jira

Probably their most popular product, Jira is used for teams to create “tickets” which represent a task, feature, bug or release. Using labels and a sleek interface, Jira provides the interface to communicate cross-collaboratively amongst teams. Some of the use cases that would warrant use for Jira would be:

* Improving the Agile software development process
* Project management
* Bug Tracking
* Software Development
* Process Management
* Workflow approvals
* Incident response and tracking

A long with those specific use cases that would warrant Jira in the workplace, Jira also offers a multitude of integrations for improved communication channels.

* Confluence (Jira’s documentation SaaS) (more on this later)
* Bitbucket (Jira’s remote version control repo hub) (more on this later)
* Slack (instant notifications)
* GitHub
* Zoom (video conferencing software)

To host Jira, there are a few different options. Some organizations choose to host on-site using their own hardware infrastructure, others opt for Jira’s cloud service. Of course, an option is to always use cloud infrastructure as well such as Azure, GCP or AWS to provide the cloud infrastructure needed to host the software.

Licensing for Jira comes in a variety of flavors dependent on the type of hosting and product features your organization needs. It’s considered to be enterprise grade software so developing a feature rich platform for dozens or hundreds of users might accrue some unneeded expenses so cost analysis would be critical prior to choosing the license that’s right for your organization.

Jira terminology

Issues

An “issue” in Jira talk refers to a single work item of any type or size that needs to be tracked for end-to-end completion. Whether it’s a feature being developed, a to-do item for marketing, or a contract that needs to be written by the legal department, “issues” are used in Jira to create transparency of an item needed for completion.

Projects

Projects are a collection of issues representing a purpose or specific goal that needs to be achieved. Issues are attached to a project and can be configured in a variety of ways. Projects are highly customizable workspaces for teams to cross collaborate. It allows for the grouping of issues by any team, purpose or intent. For example, if you group an issue by team you could have a marketing project, a development project, and a legal project. All these projects would track ongoing work for those teams.

Workflows

Workflows represent a sequential path an issue may take from the creation to completion stage. A basic Workflow may look something like this:

* A ticket is opened by a product manager to represent an Issue
* A developer picks the ticket up and changes the status to ‘In Progress’
* Once the developer completes the task, the status is then updated and tagged as ‘Under Review’. This informs a QA analyst of the work completed and that their attention is needed.
* After QA qualifies it, they’ll either send it back to the developer or change the status of the ticket to ‘Done’

Confluence

When working in an ever-changing environment, documenting new practices, changes and other need-to-know information is quite critical for an efficient product to sustain a long trajectory. Code bases will typically change many hands, service will be refined, and architecture may evolve throughout time. Confluence is there for teams to utilize as a single source of truth for any changes that need documenting.

Atlassian offers Confluence as a way for organizations to create a knowledge base or wikis to keep every on track when it comes to implementations, feature or product changes. It also serves new developers quite well when they’re onboarded. Having the knowledge base available will really expedite the process for a new employee to familiarize themselves with the products they’ll be working on.

Bitbucket

Bitbucket is Atlassian’s answer to GitHub. It works just the same, only the UI is different and offers additional features for DevOps integration. To integrate with their build platform Bamboo, it’s super plug and play. Atlassian makes integrating Bitbucket and Bamboo as easy as possible. They offer configurations for project repositories, 3rd party integrations, Jira integration and great ability to diff code with their diff views.

Merge checks, code searches, and other settings for peer review approvals when a developer makes a PR are standard as usual with this cloud-based Git client.

Bamboo

Where Bamboo might have an upper hand if it were compared with one of the more common industry standard build pipelines like Jenkins, it would be its ability to bring in built-in support for many things that an engineer would want in a pipeline out of the box.

Git branching strategies, deployment projects, software integration, test automation and no-nonsense enterprise grade permission systems. It offers a very feature rich set built right into the platform.

Version Control

If we’re talking about version control systems again, let’s quickly summarize that Git is not the only VCS out there. In the field you’ll commonly run in to either Git or Subversion otherwise known as SVN. SVN is a bit older and predates Git. The concepts and terminologies are pretty much the same.

For remote repositories there are several providers to choose from. Most notably, you’ll run into Bitbucket, GitLab, Beanstalk or AWS Code Commit. Mercurial is another free and distributed VCS providing a simple interface. It offers the same workflow necessities such as history tracking, access control etc.

The take away from this course is to be aware of the terminologies used when speaking with developers on the field. Know the common processes and workflows implemented for VCS strategies. If your familiar with terms like commits, merges, pull requests and the concepts of tracking software changes, you’ll have the power to quickly familiarize yourself with any VCS out there.