



Several Clouds

Infrastructure as Code (IaC)

Prepared for

Faculty of Mathematics and Informatics (FMI)



What is IaC

- ability to provision and support your computing infrastructure using code instead of manual processes and settings
- Any application environment requires many infrastructure components like operating systems, database connections, and storage. Developers have to regularly set up, update, and maintain the infrastructure to develop, test, and deploy applications.
- Manual infrastructure management is time-consuming and prone to error—especially when you manage applications at scale.
- Infrastructure as code lets you define your infrastructure's desired state without including all the steps to get to that state



Several Clouds

Example infrastructure

1. Network connectivity
2. Compute resources
3. Database

- Traditionally, system administrators used a combination of scripts and manual processes to set up infrastructure environments.
- Complex and time-consuming.



Configuration Management vs Provisioning

- Chef, Puppet, and Ansible are all configuration management tools, which means they are designed to install and manage software on existing servers.
- CloudFormation, Pulumi, and Terraform are provisioning tools, which means they are designed to provision the servers themselves

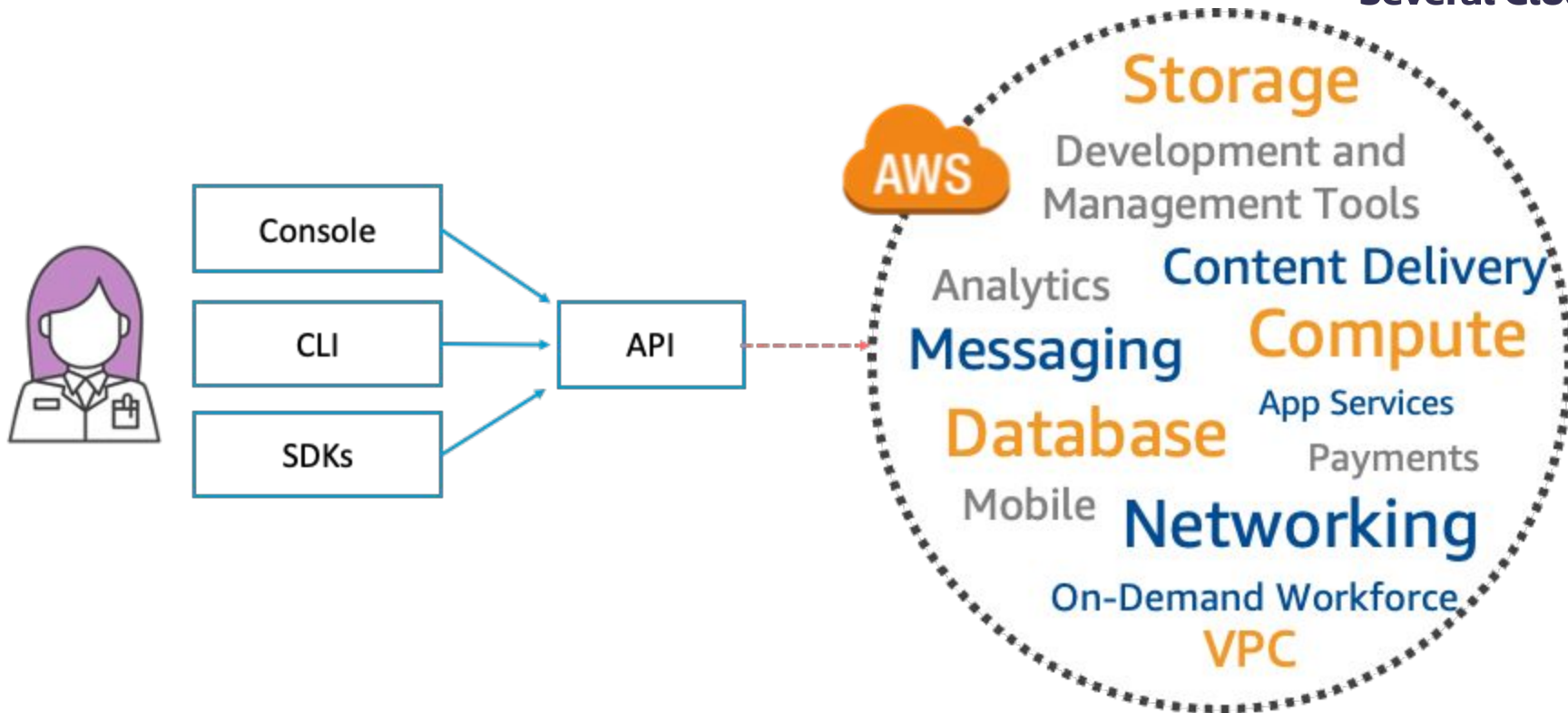


Several Clouds





Several Clouds





laC

- Infrastructure as code (laC) is used for infrastructure automation to create environments.
- The most common use of laC is in software development to build, test, and deploy applications.



Several Clouds

Advantages of IaC

- Security
- Auditability and Traceability
- Repeatability and Consistency
- Documented
- Part of the CI/CD Process
- Cost-Efficient



Approaches to IaC

manage infrastructure in a codified, repeatable way.

- Declarative

- the end state of a desired system.

```
resource "aws_instance" "example" {  
  count      = 10  
  ami       = "ami-0fb653ca2d3203ac1"  
  instance_type = "t2.micro"  
}
```

- Procedural

- all the steps to set up the resources and get to the desired system and running state

```
- ec2:  
  count: 10  
  image: ami-0fb653ca2d3203ac1  
  instance_type: t2.micro
```



Several Clouds

General-purpose language versus domain-specific language

- Chef and Pulumi, CDK allow you to use a general-purpose programming language (GPL)
- Terraform, Puppet, Ansible, CloudFormation each use a domain-specific language (DSL)



Several Clouds

When not to use infrastructure as code

- Take some time to write the code/script logic
- POC
- apply it more than 1 or 2 times



Several Clouds

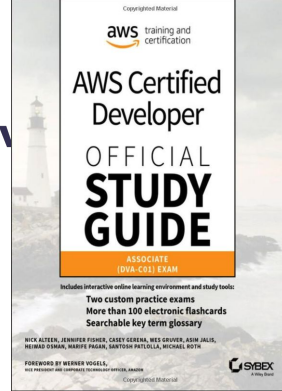
Terraform

<https://developer.hashicorp.com/terraform/tutorials/aws-get-started>

Resources

- <https://aws.amazon.com/what-is/cloud-native/>
- <https://awsstash.com/?search=%22serverless%22>
- <https://aws.amazon.com/serverless-workshops/>
- <https://aws.amazon.com/certification/certified-developer-associate/>
- <https://aws.amazon.com/blogs/architecture/lets-architect-serverless-architecture-on-aws/>
- <https://reinvent.awsevents.com/>
- <https://www.coursera.org/specializations/aws-nodejs-serverless-development>

Sev





Several Clouds

Thank you!

daniel@severalclouds.com

<https://www.linkedin.com/in/danielrankov/>

<https://www.meetup.com/aws-bulgaria/>